Adaptive Medium Access Control for Heterogeneous Wireless Sensor Networks

Giorgio Corbellini

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Giorgio Corbellini

Thèse dirigée par **Andrzej Duda et Emilio Calvanese Strinati**

préparée au sein **UMR 5217 - LIG - Laboratoire d'Informatique de Grenoble**
et de **l'École doctorale Mathematiques, Sciences et Technologies de l'Information, Informatique**

Adaptive Medium Access Control for Heterogeneous Wireless Sensor Networks

Thèse soutenue publiquement le **12 Juin 2012**,
devant le jury composé de :

Prof. Vivien Quéma
Grenoble INP–Ensimag, Président
Prof. Isabelle Guérin-Lassous
ENS Lyon , Rapporteur
Prof. Fabrice Valois
Insa Lyon, Rapporteur
Prof. Maria-Gabriella Di Benedetto
Université de Rome "La Sapienza", Examinatrice
Prof. Andrzej Duda
Grenoble INP–Ensimag, Directeur de thèse
Dr. Emilio Calvanese Strinati
CEA–LETI Grenoble, Co-Directeur de thèse

Abstract

In this PhD thesis, we consider heterogeneous Wireless Sensor Networks (WSNs) in which several sensing devices with different characteristics coexist. In contrast to a homogeneous sensor network, in heterogeneous networks different sensors may sense different physical phenomena generating traffic that have different characteristics such as monitoring temperature, pressure, and humidity. Moreover, also information criticality can be heterogeneous. Deployment of nodes also introduces heterogeneity in the network. Depending on the specific application in fact, initial deployment can be random with the result that the distribution of nodes across the playground may be non-homogeneous. In addition to this, other factors such as node death because of energy resource exhaustion, mobility, or generic fault influence the heterogeneity of the distribution of nodes.

All these characteristics can be considered as sources of heterogeneity of a WSN. Heterogeneity conditions may evolve during time and space, therefore, the design of a heterogeneous sensor networks requires adaptive mechanisms able to react to different characteristics, which is difficult to achieve. The goal of the thesis is to investigate the problems related to heterogeneity of sources in WSNs to design adaptive MAC methods that are able to take into account heterogeneity variations and are energy-efficient. We focus on two sources of heterogeneity.

First, we study the problem of multiple traffic sources with different characteristics and constraints. Providing differentiated Quality of Service (QoS) such as low latency and high delivery ratio in large and multi-hops networks is a challenge due to limited energy resources of nodes. To solve this problem we propose an adaptive MAC protocol based on the asynchronous preamble sampling (PS) approach, a simple energy saving MAC technique, coupled with the idea of using a rendezvous time for data transmission. The proposed protocol (Low-Latency MAC, LA-MAC), is able to ensure efficient message forwarding throughout a multi-hop network thanks to the transmission of bursts. When messages need to be forwarded, each receiver behaves like coordinator to organize efficient transmission of contending senders. The innovation of the proposed protocol comes from combining enriched PS preambles to locally organize data transmission in a collision limited way. Extensive numerical simulations show that LA-MAC outperforms other state-of-the-art protocols in terms of latency, delivery ratio, and energy consumption.

The precise evaluation of the energy consumption in large wireless sensor networks that use preamble sampling MAC is difficult. In this thesis, we propose an analytic model for energy evaluation of PS that depends on the instantaneous traffic load of localized regions so that it is independent of the network traffic patterns that can also be heterogeneous.

Second, we study dynamic WSNs with density of nodes that varies across space and time. Such networks are characterized by high variability in terms of node densities. Rapid density variation may affect the network state effecting the collision probability and energy consumption of devices. We address the case of dynamic wireless sensor networks in which nodes and/or radio links may appear and disappear over time due to battery exhaustion, node mobility, or network management operations. With the work presented in this thesis we show that it is possible to provide QoS support in dynamic networks using an adaptive Density Aware MAC (DA-MAC) method. The proposed protocol offers a configurable channel sensing phase during which nodes request transmission opportunity in a way that avoids collisions. With DA-MAC, nodes periodically adapt their local protocol parameters to access the channel without collisions depending on local density state. The efficiency of the proposed protocol with respect to other state-of-the-art protocols is shown with extensive numerical simulations.

Keywords

Medium Access Control (MAC), Wireless Sensor Networks (WSN), Heterogeneity, Quality of Service (QoS), density, low-power, energy evaluation, preamble sampling.

Abstract in French

Ce mémoire de thèse s'intéresse aux réseaux hétérogènes de capteurs sans fil (Wireless Sensor Networks - WSNs) constitués par une multitude de dispositifs de détection qui coexistent malgré leurs caractéristiques différentes. Contrairement aux réseaux homogènes de capteurs, chaque capteur d'un réseau hétérogène est capable de détecter et mesurer différents phénomènes physiques (température, pression, humidité) et générer ainsi un trafic avec des caractéristiques spécifiques, différentes d'un capteur à l'autre.

En effet, selon l'application visée, le déploiement initial des nœuds peut être aléatoire, résultant en une répartition non-homogène des nœuds dans l'environnement. Autres facteurs comme l'extinction d'un nœud suite à l'épuisement de sa batterie ou à une faute générique peuvent impacter l'hétérogénéité de la répartition des nœuds. Tous ces phénomènes peuvent être perçus comme une source supplémentaire d'hétérogénéité dans les réseaux de capteurs sans fil. Puisque les facteurs d'hétérogénéité peuvent évoluer tant au cours du temps que dans l'espace, il est indispensable de concevoir des mécanismes adaptatifs pour les réseaux hétérogènes de capteurs afin de réagir et de s'adapter à la dynamique du réseau. De tels mécanismes adaptatifs sont toutefois difficiles à mettre en place. L'objectif majeur de cette thèse est d'étudier les problèmes liés à l'hétérogénéité dans les réseaux de capteurs sans fil afin de concevoir des méthodes de contrôle de l'accès au canal (Medium Access Control - MAC) qui s'adaptent à la dynamique de l'hétérogénéité tout en étant économe d'un point de vue énergétique. Deux sources d'hétérogénéité sont envisagées.

Dans un premier temps, nous considérons les problématiques liées aux sources dans trafics multiples chacune dotée de caractéristiques et contraintes spécifiques. Pour pallier ce problème, un protocole MAC adaptatif basé sur une approche asynchrone est proposé ; il consiste en une méthode MAC de préservation de l'énergie, couplée à l'utilisation d'un instant de rendez-vous pour la transmission des données. Le protocole proposé, LA-MAC pour Low-Latency MAC, permet de garantir de façon efficace le transport de messages au travers d'un réseau multi-sauts grâce à la transmission d'agrégats de données (bursts). De vastes campagnes de simulations numériques corroborent la supériorité de LA-MAC en termes de latence, de taux de paquets correctement délivrés et de consommation énergétique par rapport à d'autres protocoles présentés dans l'état de l'art.

Dans un second temps, nous étudions des réseaux dynamiques de capteurs sans fil, dont la densité de nœuds varie en temps et en espace. Cette densité des nœuds dans le réseau peut se définir comme étant le nombre de dispositifs avec des données à émettre par mètre carré. En effet, de brusques augmentations de la densité résultent en un accroissement du taux de paquets perdus en raison d'une hausse de la probabilité de collision des trames. En outre, une baisse de la densité des nœuds peut causer un gaspillage énergétique dû à une écoute oisive. Dans ce mémoire, nous traitons des réseaux dynamiques de capteurs sans fils dans lesquels les nœuds et les liens radio entre ces nœuds peuvent apparaître ou disparaître au cours du temps en raison de l'épuisement de leurs batterie, ou de toute autre opération d'administration du réseau, comme par exemple le déploiement de nœuds additionnels. Le travail présenté démontre qu'il est possible de fournir un support à la qualité de service (QoS) dans les réseaux dynamiques grâce à une méthode MAC adaptative et consciente de la densité, baptisée DA-MAC pour Density Aware MAC. Avec DA-MAC, les nœuds s'appuient sur la valeur de la densité locale et adaptent périodiquement les paramètres locaux qui régissent le protocole afin d'accéder au canal sans collision. L'efficacité du protocole proposé est présentée en comparaison d'autres protocoles de l'état de l'art dans de vastes campagnes de simulations numériques.

Table des matières

Table des figures

Liste des tableaux

Notation

Abbreviations and Acronyms

ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
CAP	Contention Access Period
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CH	Cluster Head
CP	Contention Period
CR	Contention Round
CTI	Cross Technology Interference
CW	Contention Window
DCF	Distributed Coordination Function
FEC	Forward Error Correction
FDMA	Frequency Division Multiple Access
FRTS	Future Request To Send
FFD	Full Function Device
GTS	Guaranteed Time Slot
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
MAC	Medium Access Control
M2M	Machine to Machine
NIC	Network Interface Card
NLOS	Non Line Of Sight
PGR	Packet Generation Rate
RFD	Reduced Function Device
RPL	Routing Protocol for Low power and Lossy Networks
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
VC	Virtual Cluster
VoIP	Voice over Internet Protocol
WiFi	‘Wireless Fidelity’
WSN	Wireless Sensor Network

Thesis Specific Notation

The following list is not exhaustive and consists of the most relevant notation used in the dissertation. Vectors are denoted by underlined letters (e.g. $\underline{P}$).

Other notational conventions are summarized as follows :

Notation applicable in all chapters

C^i	Children of node i
$\mathscr{C}_i$	Source s_i and destination d_i
D	Density of nodes : the average number of neighbors per node
d_i	Destination of pair $\mathscr{C}_i$
E_l	Energy consumption of channel polling radio state
E_r	Energy consumption of reception radio state
E_s	Energy consumption of sleeping radio state
E_t	Energy consumption of transmission radio state
$\mathbb{E}\{x\}$	Expectation of variable x
γ_i	SNR between source s_i and destination d_i
Γ	Probability of generating an alarm burst
L_{dB}	Path loss attenuation [dB]
N	Network population
N_0	Noise variance
P_l	Power consumption of channel polling radio state
P_r	Power consumption of reception radio state
P_s	Power consumption of sleeping radio state
P_t	Power consumption of transmission radio state
$P_{\max}$	Maximal transmit power
$\mathbb{P}\{x\}$	Probability of event x
r_m	Packet Generation Rate
R	Transmission rate
$\mathbb{R}^+$	Set of all positive real numbers
s_i	Source of pair $\mathscr{C}_i$
$S_i(t)$	Priority of traffic source i over time
t_d	Time spent in transmission radio state (transmission of data frames)
t_f	Duration of a protocol frame
t_l	Time spent in channel polling radio state
t_p	Duration of a preamble message
t_r	Time spent in reception radio state
Tr_i	Traffic type i
t_s	Time spent in sleeping radio state
t_t	Time spent in transmission radio state
t^i_{wp}	Wake-up instant of parent node i

Chapter 1

Introduction

1.1 Background and Motivations

Wireless Sensor Networks (WSNs) are used in many different domains and applications. During the last decade, Academia and industry have been working on improving the efficiency of sensor components such as battery capacity, chip design, and radio equipment. The thrust results in an increasingly wider application of WSN in many fields : unobtrusive habitat monitoring [1, 2] of wildlife [3], tracking of goods and objects [4], ad hoc deployments for disaster management and precise agriculture [5].

Military uses of sensor networks are also very frequent for example for battlefield surveillance, vehicular traffic monitoring or enemy position tracking. The strength of wireless sensors is that they allow observing the physical world at a granularity level which was "unperceived before" with reduced costs [6].

In the context with a wide variety of applications and interests, effective protocols for routing data, accessing the media and signaling are always needed. Therefore, provided that each application is specific with its peculiar requirements and characteristics, it is difficult to discuss generic application requirements or specific research directions for wireless sensor networks.

Wireless sensor networks must operate unattended in an autonomous way for a long time. Most of the existing WSN run with battery supplied nodes and in many cases battery replacement or re-charge is not possible. Scarce energy resource may result in a short lifetime, so that energy management that minimizes energy consumption is crucial.

Radio equipment is the most consuming element of a sensor node, for this reason the efficient use of the radio is mandatory in WSNs [7]. Link layer operations such as Medium Access Control (MAC) govern the use of the radio equipment, determining when it must be turned on and when the node can remain in sleep mode. As a result, along with organization of multiple access, MAC methods for WSNs must be energy efficient.

Wireless sensor networks consist of several individual devices that are tiny and resource limited in terms of computation, communication, storage, and power supply. Nevertheless, a WSN considered as an overall entity that interconnects individual devices, is quite powerful and permits to sense physical phenomena across regions that may be very large and difficult to reach such as mountains or forests. The result is that WSNs typically present dense to very dense distribution of nodes, and it is extremely challenging to make all these low complexity devices operate as coherent and efficient systems.

In Homogeneous-WSNs, sensor devices all have the same characteristics such as energy consumption, processing capacity, and radio equipment. In contrast, if the devices that co-exist in WSNs have different characteristics, we refer to a sensor network as a Heterogeneous-WSN. Moreover, different sensors may sense different physical phenomena generating traffic that have different characteristics such as monitoring temperature, pressure, and humidity, thereby introducing different reading rates at the sensors. Also information criticality can be heterogeneous. All these characteristics can be considered as sources of heterogeneity of a WSN.

The design of a heterogeneous sensor networks requires adaptive mechanisms able to react to different characteristics, which is difficult to achieve. Limited processing resources and strong energy consumption constraints require MAC methods for WSNs to be simple and energy conserving.

The goal of the thesis is to investigate the problems related to heterogeneity of sources in WSNs to design adaptive MAC methods that are energy-efficient and able to take into account heterogeneity. We focus on two sources of heterogeneity. First, we study the problem of multiple traffic sources with different characteristics and constraints. Second, we study dynamic WSNs with density of nodes that varies across space and time.

1.2 Thesis Outline

The outline of the thesis is as follows :

Chapter 2 - Medium Access Control in Wireless Sensor Networks. Many are MAC protocols for WSN present in the literature. In this chapter, we focus on presenting the most relevant state-of-the-art protocols to provide a comprehensive survey of energy efficient methods. Existing solutions are organized in three families depending upon network organization, *schedule based*, *synchronous duty cycle*, and *random access* protocols. For each of the considered protocol, we discuss its main characteristics, advantages, and drawbacks.

Chapter 3 - Low-Latency MAC for Heterogeneous Wireless Sensor Networks : LA-MAC. In this chapter, we face the problem of supporting multiple heterogeneous traffic in WSNs. In large and dense multi-hops sensor networks, providing QoS such as low latency and high delivery ratio is a challenge due to limited energy resources of nodes. We came out with a novel MAC protocol, “LA-MAC”, able to adapt its duty cycle to react to load fluctuations. LA-MAC, based on the asynchronous preamble sampling (PS) approach, is able to ensure efficient message forwarding throughout a multi-hop network thanks to the transmission of bursts. The proposed protocol is the result of a deep investigation of the PS approach ; advantages such as energy savings and simplicity are coupled with the idea of fixing a *rendezvous* for data transmission. The innovation of the proposed protocol comes from combining enriched PS preambles to locally organize data transmission in a collision limited way. LA-MAC results in lower latency and higher delivery ratio than B-MAC and X-MAC in several multi-traffic scenarios.

Chapter 4 - Energy Analysis of Preamble Sampling Based MAC. Wireless sensor networks are battery supplied and battery replacement or re-charge is usually not possible in many cases. Therefore, a major issue in wireless sensor networks is energy efficiency of protocols to ensure long network lifetime. MAC protocols play a crucial role in fighting energy waste in WSNs because they govern the use of radio equipment, the major energy demanding part of nodes. The precise evaluation of the energy consumption in large wireless sensor networks that use preamble sampling MAC is difficult. In this chapter, we propose a novel analytic model, partially deterministic and partially probabilistic, to evaluate energy efficiency of preamble sampling MAC protocols. The model is independent of the traffic pattern of the network and is based on the instantaneous traffic load of localized regions. The model is used to analyze and compare the consumption of three protocols : B-MAC, X-MAC, and LA-MAC. The proposed approach is flexible and it can be easily used to analyze the efficiency of other preamble sampling protocols.

Chapter 5 - Density Aware MAC for Dynamic Wireless Sensor Networks : DA–MAC. Dynamic Wireless Sensor Networks (WSN) are characterized by high variability in terms of node densities. Density of nodes can be defined as the number of active nodes (i.e. nodes that need to transmit or relay data packets) per square meter. Rapid density variation may affect the network state. Fast density increase results in higher packet loss due to higher probability of frame collisions. Density reduction may lead to energy waste due to idle listening (energy is wasted in listening to the channel without real needs).

In this chapter, we consider *dynamic wireless sensor networks* in which nodes and/or radio links may appear and disappear over time due to battery exhaustion, propagation conditions, node mobility, or network management operations (*e.g.* deployment of additional sensors). Such network dynamics may lead to degraded performance impacting the operation of all protocol layers including MAC and routing. To provide QoS support in dynamic networks, we propose to use density awareness to govern the behavior of an adaptive MAC method. We came out with "DA-MAC", a density aware MAC able to offer a configurable channel sensing phase during which nodes request transmission opportunity in a way that avoids collisions. The proposed MAC is based on preamble sampling approach. With DA-MAC nodes periodically adapt their local protocol parameters to access the channel without collisions.

We consider that DA-MAC supports multi-hop networks with convergecast traffic towards a sink.

Chapter 6 - Conclusions and Future Work. This chapter summarizes the thesis main contributions and perspectives.

Chapter 2

Medium Access Control in Wireless Sensor Networks

The design of MAC methods for wireless sensor networks is a well investigated problem. Limited processing resources and strong energy consumption constraints, require simple and energy conserving MAC methods for WSNs. Moreover, a multitude of applications use WSNs, so MAC methods must also be adaptive and flexible. This is difficult to achieve. Existing solutions basically adopt three approaches. In the first one, important control traffic is used and nodes adopt a synchronized network-wide TDMA structure (D-MAC [8], TRAMA[9]). The second approach aims at synchronizing nodes on a common sleep/wake-up schedule. To do so, short synchronization messages are periodically exchanged between nodes (SMAC [10], TMAC [11]). In the third approach, node sleep/wake-up schedules are independent and not synchronized. Each node follows its own schedule that consists in sleeping most of the time and sensing the channel with a given periodicity. A node that has data waiting in its packet queue transmits a long preamble frame prior to sending data. The time duration of a preamble *must be long enough to cover two consecutive wake-up instants of a potential receiver (Aloha with Preamble Sampling [12], LPL (Low Power Listening) in B-MAC [13], and CSMA-MPS [14] aka X-MAC [15]). The* synchronous preamble sampling *combines the last two approaches. Nodes make use of very short preambles and require tight synchronization between each other (WiseMAC [16], Scheduled Channel Polling (SCP) [17]).*

The chapter is organized as follows. MAC methods for WSNs and their issues are introduced in Section 2.1. Three MAC approaches are described in Section 2.2. Conclusions are drawn in Section 2.4.

2.1 Introduction

Wireless sensor networks must operate unattended in an autonomous way for a long time. Devices are usually battery supplied and typical lifetime requirement for WSNs is of the order of years. Moreover, battery re-charge or replacement being impossible in many cases, minimizing energy consumption without jeopardizing performance is the main design goal [7].

The network *lifetime* is perhaps the most important metric for the evaluation of WSNs performance, however, many definitions exist [18]. Several definitions can be found in the literature, some examples are "time until the first node exhausts its battery capacity", "time until a given area of interest is covered with alive nodes" or "the minimum time when either the percentage of alive nodes or the size of the largest connected component of the network drop below a specified threshold".

Independently of a specific definition, the lifetime concerns energy efficiency of a network that is a consequence of both energy efficiency of single devices and efficiency of the protocols that they use to interact and communicate. In this work, we focus on the energy efficiency of communication and interaction of devices, in particular we focus on efficiency of MAC *i.e.* if devices waste energy because of ineffective use of their radio. If a sensor equipped with a series of AA batteries with a capacity of 3000 mAh is left in continuous transmitting mode, it would drain its current in about 4 days [19] far away from the typical requirement of years [4]. Energy consumption and efficiency of non-radio equipment of a sensor node is much smaller than the energy consumed by radio so is out of the scope of this work.

To save energy, devices aim at achieving low duty cycles : they alternate sleep periods (radio switched off) and active periods (radio switched on). As a result, the challenge of the MAC design is to synchronize the instants of receiver and sender active periods resulting in very low network duty cycle.

Along with stringent energy consumption constraints, WSNs differ from other wireless networks such as IEEE 802.11 or cellular networks, for several additional aspects. *Applications* using WSNs may be very different and diverse such as habitat monitoring [1], environmental monitoring [2], and wildlife tracking [3]. The *deployment* of sensor nodes depends on the physical phenomenon to sense, thus it may vary from deterministic and static [5] to fully random and dynamic [4]. Resulting *topologies* may have different complexities including single hop star, multi-hop chain, multi-hop trees or a graph. Moreover, topology also depends on *mobility* of sensors and sinks. The *Number of nodes* strongly depends on the specific application and it can go from few tens in habitat monitoring scenarios to hundreds or thousands in world-wide large networks [4]. As a consequence, the cost per device must be very limited. Typical *Traffic load* is light and sensed data must converge from sensors to sink units, using a convergecast traffic pattern. Moreover, *radio links* between sensor nodes are very volatile resulting in packet errors or losses due to wireless effects such as multi-user interference, fading or multi-path. Another context in which wireless sensor networks support diverse applications in various and ubiquitous scenarios, are the Machine-to-Machine (M2M) networks [20].

Therefore, many existing MACs cannot be employed in WSNs and specific WSNs-tailored MAC must be designed.

Major sources of energy dissipation in WSN are the following :

- *Idle listening.* Idle listening is the activity of listening to the channel without the reception of any air frame. The problem is typical for non synchronous systems and the time spent by a sensor node in such state leads to a waste of energy. If a node was able to know when the sender will send a frame to it, it would sleep until the instant of transmission, thus completely avoiding idle listening.

– *Overhearing*. When a node overhears a frame that is destined to someone else it wastes energy due to the activity of reception, decoding, and processing of the frame.
– *Collisions*. Collisions occur if a node receives frames from multiple senders at the same time or if two receptions overlap for some time. The result is that the receiver is not able to decode any frames that need to be re-transmitted.

2.2 Energy Efficient MAC protocols

The literature on MAC methods for WSNs is extensive [21][19]. MAC protocols for WSNs published during last 10 years follow three main approaches with respect to the network organization (cf. Fig. 2.1) ; in the first one, nodes are organized at the network level with dedicated resources allocated to each node in the domain of time, frequency, or code. In the second one, nodes share the same wake-up and sleep schedule to achieve a synchronous duty cycle. In the last one, channel access is not organized and nodes contend for transmission. Hybrid protocols complete this classification. In Fig. 2.1, protocols are represented as blocks interconnected among each others with arrows. Vertical order follows the year of publication. If an arrow goes from protocol A to B, it means that authors

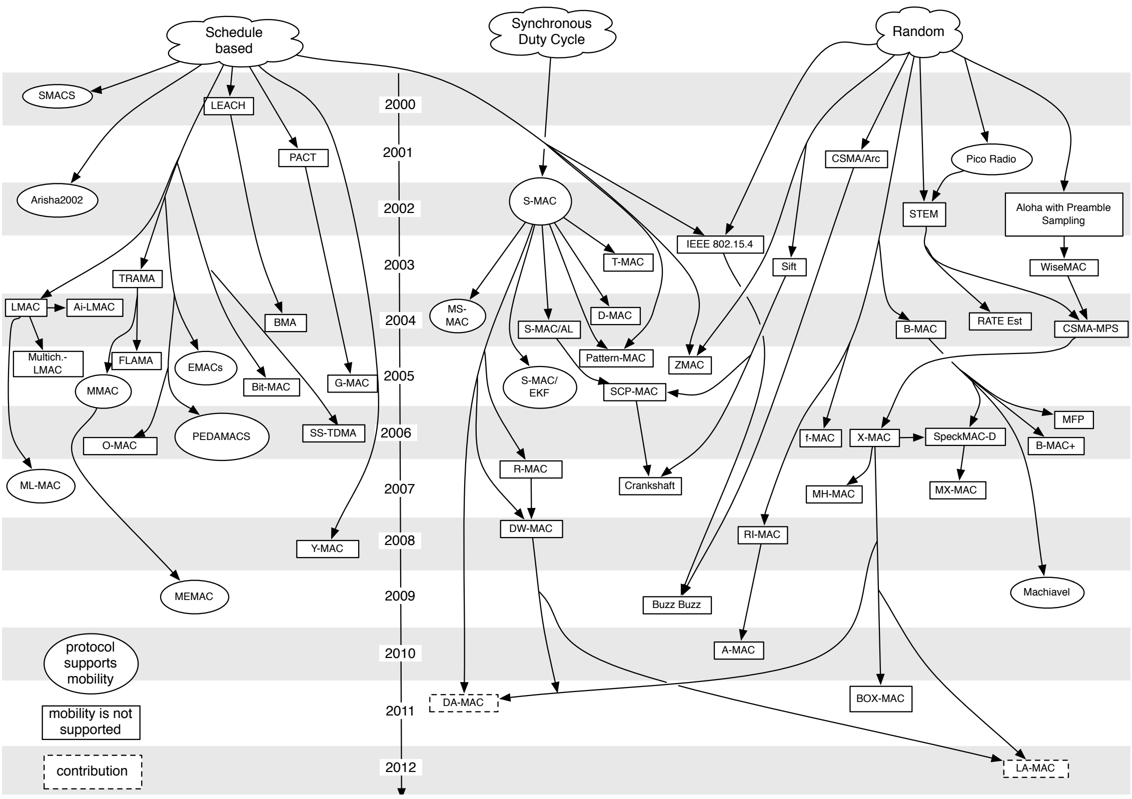


Figure 2.1 – MAC protocols for Wireless Sensor Networks.

of protocol B have been inspired by the innovation of protocol A. If a protocol name is within an oval block it means that it supports mobility to some extent, protocols in rectangular blocks, do not support mobility. From Fig. 2.1 it can be observed that some design guidelines have inspired a large number of protocols. For example, the idea of using using duty cycles and focus transmissions during active periods was first introduced in S-MAC [10]. Duty cycling nodes activity is the basics of T-MAC [11], DW-MAC [22], Crankshaft [23] and other protocol that are investigated in this chapter. STEM [24] is a precursor of preamble sampling protocols, senders transmit advertisement packets on a separate radio to alert the receivers of an incoming transmission ; the same idea, with a single radio, is used several protocols such as B-MAC [13], CSMA-MPS [14], or MX-MAC [25].

2.2.1 Schedule-based Protocols

The main advantage of schedule-based protocols is the absence of collisions that results from the network-wide organization of nodes. The problems of *idle listening* and *overhearing* are also solved and as a consequence, the energy waste is limited. However, the cost of organizing channel access of nodes at the network level may be prohibitive in WSNs with stringent energy constraints.

Schedule-based protocols for WSNs consist in allocating a piece of channel access to each node, on a permanent basis. Such piece of channel access can be a time slot in TDMA-like (Time Division Multiple Access) protocols, a frequency band in FDMA-like (Frequency Division Multiple Access), a code in CDMA-like (Code Division Multiple Access) protocols or a combinations of these elements.

Along with the canonical IEEE 802.15.4 MAC standard [26] we discuss the major protocols of this family.

The IEEE 802.15.4 MAC protocol supports two communication modes : beacon-enabled and non-beacon enabled. In the first mode, time is divided into supeframes. Each superframe is composed of an active period and an inactive period whose duration is adaptive. The active period is divided into two parts : a Contention Access Period (CAP) and a Contention Free Period (CFP) as depicted in Fig. 2.2. The network is organized in a hierarchical way with Full Function Devices (FFDs) that act as network coordinators and several Reduced Function Devices (RFDs). RFDs can transmit messages both during the CAP and the CFP period, depending on the decision of the coordinator. The coordinator periodically send beacons (with period $1/(\textit{Beacon Interval})$, *i.e.*, $1/BI$) to maintain the slot structure and advertising the beginning of the next superframe. The devices that receive the beacon synchronize their wake-up schedule with the next beacon transmission. In each beacon, several pieces of information are embedded such as the length of the active and inactive periods, BI, and the addresses of the nodes allocated with Guaranteed Time Slot (GTS) for a collision free transmission. If the coordinator does not reserve any GTS for any node in particular, all devices must use the CAP period for transmission, that is, they must contend for channel access using a CSMA/CA procedure explained below (cf. Sec. 2.2.3).

In the second mode, that is the non-beacon enabled mode, devices are not synchronized so that when a node wants to transmit data, it uses a different CSMA/CA procedure with respect to the beacon enabled mode, that is, without waiting for the beacon. When the sender wakes up, sets a random *back-off timer*, and listens to the channel until the *back-off timer* fires. If channel is clear during all the duration of the *back-off timer*, the sender transmits a Request to Send (RTS) the coordinator that allows the the transmission sending back a Clear to Send (CTS) message ; then data transmission can occur. Other senders that listen to the traffic of the first one, must wait until transmission end to set their own *back-off timer*. In the non-beacon enabled mode, the coordinator has to remain awake to receive possible incoming RTS and cannot save energy ; the result is that both overhead and end-to-end delay are high.

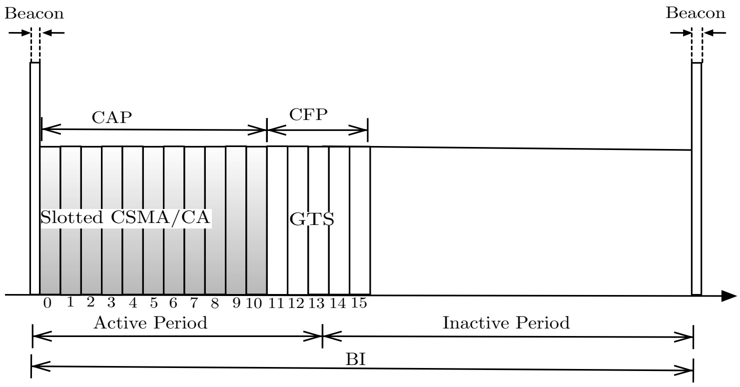


FIGURE 2.2 – Superframe structure if the IEEE 802.15.4 MAC in beacon-enabled mode.

The Low-Energy Adaptive Clustering Hierarchy (LEACH) [27] protocol is a clustering protocol that seeks to minimize energy dissipation of nodes thanks to localized coordination of clusters, cluster-head rotation and local compression of data. LEACH protocol is structured in rounds. Each round comprises a *cluster set-up phase* in which clusters are created and a *steady-state phase* in which data transmissions occur. Prior to the cluster set-up of each round, a prefixed number of nodes must be elected cluster-heads (CHs) using a probabilistic rule. During the cluster set-up phase, after that each non-CH node has decided to which cluster it belongs, it must communicate its decision to the CH using a CSMA-like scheme. During this phase, all CH nodes must keep their radios in receiving mode to collect incoming messages. Once the CHs have received all messages from nodes, they create the time schedule and broadcast it to the cluster members. Then, the steady-state phase for data transmission starts. With LEACH, inter-cluster interference is limited using different CDMA codes for intra-cluster communications. The cluster head election is not energy efficient, moreover, cluster head nodes must be able to communicate with the sink using a direct long-haul transmission, thus consuming considerable energy in large networks.

Arisha [28] is a TDMA-based Medium Access Control protocol for mobile sensor networks. In the protocol, it is assumed that the sink is able to manage the topology information about all the sensor nodes and that it coordinates the communication by allocating data slots. Two algorithms for slot assignment are defined, in both cases the allocation is related to the actual data traffic of nodes. In the first one, *breadth first search*, slots are assigned so that nodes having the same parent in the routing tree have contiguous slots. Starting from the leaf nodes upward until the sink, this allocation method results in energy savings and facilitates data aggregation. Each node in the tree in fact periodically collects all data from its children first and then it forwards them to its next hop, afterwards it goes back to sleep. The second slot allocation mode, *depth first search*, allocates slots over each route from the sensing node towards the sink. This method avoids packet drops and results in lower latency ; however, nodes turn more frequently their radio on jeopardizing energy consumption.

PEDAMACS (Power Efficient and Delay Aware MAC for Sensor Networks) [29] provides a collision-free MAC for WSNs in situations of low mobility of nodes. The protocol bases the medium access procedure on the work of a unique powerful sink that is able to perform long-haul communications, that is, to send a message directly to each node of the network. PEDAMACS protocol ensures collision free communications thanks to a TDMA mechanism in which all slots are allocated by the sink. The PEDAMACS protocol is organized in 4 phases. In the first, *topology learning phase*, the sink initiates a flooding procedure with routing packets so that all nodes have the knowledge of their neighbors. In phase 2, *topology collection*, the sink requests all nodes to report on their local topology information. When all topology packets are collected by the sink, it can run the third phase, the *scheduling phase* during which it communicates to all nodes the TDMA slots that they can use for transmission. Each device has assigned one time slot for transmission. During last phase, the *adjustment phase*, nodes communicate to the sink the recent topology modifications if some occurred. If new nodes appear or nodes move, this information is sent back from new neighbors to the sink during this phase. Moreover, if some slots were assigned to multiple nodes due to an allocation error, in this phase the sink can make small corrections. As in LEACH, the assumption that the sink can reach all nodes within its radio range, is strong and is rarely satisfied in low-power and energy constraint WSNs.

Network Organization	Protocols
Schedule-based	Ai-LMAC [30], IEEE 802.15.4 MAC [26], L-MAC [31] TRAMA [9], FLAMA [32], Mobile-LMAC [33], MEMAC [34] M-MAC [35], O-MAC [36], Arisha [28], Bit-MAC [37] PACT [38], G-MAC [39], PEDAMACS [29] BMA [40], LEACH [27] , EMACs [41], S-MACS [42] SS-TDMA [43], Y-MAC [44], Multi-channel L-MAC [45]
Synchronous Duty Cycle	S-MAC [10], MS-MAC [46], S-MAC/EKF [47], T-MAC [11] R-MAC [48],D-MAC [8], DW-MAC [22], S-MAC/AL [49]
Random Access	CSMA/ARC [50], B-MAC [13], B-MAC+ [51] X-MAC [15], STEM [24], CSMA-MPS [14] MH-MAC [52], MX-MAC [25], PicoRadio [53] BuzzBuzz [54], MFP [55], RATE EST [56] WiseMAC [16], RI-MAC [57], A-MAC [58] , Machiavel [59] BOX-MAC [60], Sift [61], SpeckMAC-D [62] f-MAC [63]
Hybrid	Crankshaft [23], IEEE 802.15.4 MAC [26], SCP-MAC [17] Z-MAC [64], P-MAC [65]

TABLE 2.1 – MAC Protocols for WSNs.

Bit-MAC [37] is a deterministic and collision-free protocol for dense wireless sensor networks. The network structure is based on a spanning tree rooted at the sink, so each node can be at the same time a parent and a child. Communication is restricted to parent-to-children and vice-versa only. Each parent defines a TDMA schedule for communicating with its children (small star-based clusters are so created). Communication is organized in rounds and in each round the parent node broadcasts a beacon specifying if in the current round transmission direction is down-link or up-link. Thanks to multiple channels, several parts of the tree can operate in parallel. The resulting protocol is very similar to IEEE 802.15.4 beacon-enabled MAC protocol [26].

The Power Aware Cluster TDMA (PACT) [38] protocol exploits the high density of nodes and passive clustering to create a sub-network with cluster-heads and gateways for inter-cluster communication. The role of cluster-heads and gateways is assumed by nodes that have an adequate energy level and is rotated during time. Each TDMA frame consists of a control part and a data part. In the control part, each node has a reserved control slot that is used to announce the destination of upcoming data. All nodes must be awaken during this period, so that only nodes that have data to send or receive will be awaken during the following data part of the frame. Passive clustering is a fully distributed clustering technique according to which clusters are created without the need of sending explicit control messages. Nodes form the clusters by overhearing packets of neighbors. Once clusters are created, *clustering status information* is sent in the control part of each TDMA frame. Passive clustering results in low overhead for cluster building, however the global protocol overhead is high because of the control part of TDMA frame.

Similarly to PACT, the Gateway-MAC (G-MAC) [39] protocol provides a cluster-based TDMA mechanism that eliminates the intra-cluster idle listening resulting in large energy savings. In G-MAC, the access scheme is managed by a cluster head node (the gateway) that is self-elected thanks to a distributed rule based on the residual energy state of nodes. All nodes are equal, so after roll-out they all have the same probability of self-electing as gateway. If a *gateway election collision* occurs, nodes wait a prefixed amount of time and then restart the self-election procedure. Once a node becomes the gateway, it maintains the role until its energy state does not change, at this time it broadcasts the request for a new gateway election. G-MAC operation consists of three periods : during the first one, *collection period*, the gateway collects Future-RTS (FRTS) messages from all nodes belonging to its cluster and RTS messages coming from gateways of other clusters. All messages are processed by the gateway that computes the TDMA slot schedule. During the second period, the *traffic indication period*, all nodes in the cluster must wake up and listen to the Gateway Traffic Indication Message (GTIM). GTIM contains the effective time schedule and the possible request for new gateway election. During the last period, *distribution period*, nodes use their slots for collision-free local communications. Nodes that have any data to send or receive can go to sleep until the next frame starts. The main drawback of G-MAC is in the large overhead introduced by the collection period.

Along with LEACH, PACT, and G-MAC, the Bit-Map-Assisted (BMA) [40] protocol addresses the problems of intra-cluster communications in a wide network composed of several clusters, each one with low-to-medium traffic load. BMA procedure is organized in rounds and each round comprises two phases : a *cluster set-up phase* for cluster head election and a *steady-state phase* during which nodes use TDMA slots for transmission.

L-MAC [31], is a self-organizing TDMA access scheme that guarantees collision-free data transmission. The network is composed of gateways and sensing devices ; each sensor synchronizes its clock with respect to the closest gateway. Time is organized in frames and slots. Each node *controls* one slot per fixed-length frame during which it may send data to any of its neighbors. Slots are re-used, *i.e.* the same slots can be adopted by distant nodes (at least three hops away). In L-MAC, slots assignment is accomplished using a distributed algorithm. Each gateway broadcasts the information about which slots are still available in its frame. After the duration of one frame, each node knows all the available slots in its range. Then nodes randomly choose one slot among the available, if there is a collision of slots, devices wait a back-off time before choosing another slot. A given slot is considered *controlled* by a node if all its neighbors agree with this. Thus, all nodes must store a slot allocation table relative to their complete neighborhood. A given slot remains controlled by the same node until its battery exhaust or it has no more data to send. The major drawback in L-MAC is scalability : in dense networks, some nodes may be unable to transmit because all slots are already assigned. This problem is solved in Multi-channel L-MAC [45] where slots are re-allocated in different frequency bands.

The self-Stabilizing MAC (SS-TDMA) [43] protocol is a TDMA access scheme working with rectangular grid networks running convergecast and local gossip communications. Depending on the pattern of traffic that must be supported by the network, the slot allocation procedure of SS-TDMA slightly changes in terms of how frequently nodes can transmit to reduce collisions. SS-TDMA uses a fixed schedule that remains valid for the entire lifetime of the network.

Ai-LMAC [30], the adaptive information-LMAC protocol, improves L-MAC by allowing nodes to control more than one slot of the TDMA frame. As in L-MAC, slot assignment follows a distributed rule (no coordinating node is needed) with the difference that nodes are structured in a parent-child hierarchy so that parent nodes can *suggest* to their children how many slots they should claim for. Ai-LMAC is adaptive with respect to traffic load, *i.e.*, slots are assigned with a higher priority to those nodes that have large amount of data to send. In Ai-LMAC, each parent node stores the information about the amount of data already transmitted by its children in Data Distribution Tables (DDTs). Thanks to DDTs, a parent node may roughly estimates how much data can be expected from a given child to suggest the adequate number of slots to ask.

Mobile-LMAC [33] introduces mobility into the L-MAC protocol. In mobile environments, it may happen that a node has no gateway within its neighborhood ; in such a case, the node itself can start the slot allocation procedure instead of waiting for a gateway to start. Each node periodically receives from its neighbors control packets including a *bit-masks* with current slot allocation. If no control packet is received during the last period, a node consider itself in movement and it must be assigned a new slot. If several nodes start the slot allocation procedure at the same time, the *Age of Synchronization* field of the control packets is compared and the oldest node has higher priority. In contrast with L-MAC, slot allocation in M-LMAC is not permanent, *i.e.*, slots can be released if unnecessary ; moreover, in case of slot collision one node can decide to free its slot to choose a new one.

M-LMAC is a TDMA-based MAC and thus it is collision free ; however, two major drawbacks are present : first the protocol is not-easily scalable because of the limited size of the bit-masks. Second, in case of highly mobile networks, the information contained inside bit-masks becomes inconsistent.

The Self-organizing MAC for Sensor Networks (S-MACS) [42] protocol schedules the access of nodes by adopting both a temporal and frequency scheme. Sensor nodes are assumed able to tune their carrier frequency to differentiate links on a frequency basis. After initial network deployment, S-MACS uses a self-organizing setup procedure during which communication links are established between nodes and collisions are reduced by the use of different carrier frequencies for each link. Synchronization of nodes is not necessary ; when nodes initially wake up at a random time, they immediately start listening to the channel on a prefixed band for a random period. Then, when nodes discover each other, they can establish a permanent communication link by choosing a random carrier frequency within the large number of available bands. Once a transmission link is created, nodes can communicate using a TDMA scheme. In S-MACS almost all network nodes are supposed static with a minority of mobile nodes. Stationary nodes periodically broadcast control messages that are eavesdropped by mobile nodes. All mobile nodes keep track of all stationary neighbor control messages within a registry containing useful information such as RSSI, node address, and SNR. Connections are established and released by the mobile node itself according to the location and mobility information inferred from the local register. Although simple to implement, S-MACS is not energy efficient.

In TRAMA [9], devices control time slots depending on their needs. The TRaffic-Adaptive Medium Access protocol is a TDMA scheme based on a distributed slot selection algorithm that requires nodes to exchange two-hop information, but only with low frequency to reduce the signaling overhead. When running out of packets, nodes may release their slots. The time frame is divided into two periods. A CAP for signaling and a CFP to transmit data. The protocol consists of two phases, during the first one, nodes exchange their slot requests also specifying the identity of their destinations and elaborate requests of neighbors. During the second phase, each node knows when it must wake up for data transmission or reception.

In the FLow-Aware Medium Access (FLAMA) [32] protocol, nodes are assigned some “weights" depending on the number of incoming and outgoing flows that they have. The larger is the weight, the greater the probability of having multiple slots assigned for that node. The time frame of FLAMA is organized in *random access periods* and *scheduled access periods* like in TRAMA protocol. However, in FLAMA, nodes do not need to periodically exchange information between two-hop neighbors.

The Y-MAC [44] protocol aims at reducing latency and collisions in case of heavy traffic load with a multi-channel MAC. Y-MAC structures network communications in a TDMA fashion in which the superframe is divided into a *broadcast period* and a *unicast period* both composed of several communication slots. Communication slots are assigned to nodes according to a distributed technique like in L-MAC and are released once a node runs out of battery. Nodes synchronize with the sink and each one has a slot allocated for reception. Each node knows the reception slots of neighbors. The total number of available slots in the network must be fixed prior to network roll-out according to the estimated node density. When a node needs to send unicast messages, it uses the reception slot of destination to send collision-free messages using a *base channel*.

In conditions of heavy traffic load, the multi-channel mechanisms is enabled : after the initial transmission on the base channel, successive packets are sent each on a different channel following a prefixed frequency hopping sequence. As a result, collisions are avoided and latency is reduced thanks to parallel transmissions.

In the Energy-efficient MAC (EMACs) [41] protocol, the time frame is divided into several time slots, and each node can control only one slot during which it transfers data in a collision-free manner. Slots are not assigned by a controller or a base station, but nodes themselves can do their choice. Nodes can have three operational modes : *active*, *passive*, or *dormant*.

When a node is in *active* mode it accepts messages from *passive* nodes. When nodes are in *passive*, mode they conserve energy by keeping track of only one *active* node and use it like a sort of a gateway node to forward their messages. *Dormant* nodes only turn their radios on low-power mode (for example when a node has its battery drained).

Each time slot is divided into three parts : Communication Request (CR), Traffic Control (TC), and the data section. In the CR section, the *active* node that controls the slot listens to *passive* node requests (like RTS messages in IEEE 802.11 Distributed Coordination Function (DCF) [66]). Nodes that do not have a request for the current slot owner, keep their transceiver in a low power state during the entire CR section. In the TC part, the owner of a slot always transmits a TC message (independently of the fact that it has data to send), because this represents the heart beat of the network and it is used by new nodes (those that have recently woken up) to synchronize themselves to the network. All nodes sleep during non assigned slots. The time slot controller uses a TC message to acknowledge the *passive* node requests and send the schedule for data transmission of *passive* nodes. If a node is not the intended receiver (its ID is not included in the TC message) nor its request was approved, it goes to sleep. After the TC message, data transfer can start. *Passive* nodes attach themselves to a single *active* node and only communicate with it. After data collection from *passive* nodes, the *active* one needs to forward messages. To do so it announces its necessity to transmit during its TC slot. All *active* nodes listen to all their neighboring *active* nodes so that messages can be forwarded during this period. Although node roles rotate for reducing energy consumption, role rotation is not optimized resulting in non-homogeneous energy consumption of nodes.

M-MAC [35] is a modified version of the TRAMA Protocol supporting mobile nodes. The innovation of M-MAC consists in adapting the frame duration with respect to node mobility : the higher estimated mobility within 2-hops neighbors, the shorter the frame duration. Each node is supposed aware of its current position and is able to estimate its future position thanks to a auto-regressive model of the first order (AR-1) [67]. Nodes periodically estimate their future position and broadcast the mobility information inside the header of each MAC packet sent in the CAP part of the frame (the frame structure is the same as in TRAMA). In this way, each node can collect mobility information of its 2-hops neighbors and adapt duration of CAP and CFP accordingly. In M-MAC, nodes are organized in clusters and cluster head election rule depends on the residual energy state of nodes [68] ; the last slot of CFP is reserved by the cluster head for broadcasting all received mobility information to cluster members. The frame duration is always the same for all nodes of the network and its size is modified only during each Global Synchronization Period (GSP). Between two GSP periods only CAP and CFP duration are adapted depending on the degree of mobility. The assumption of position awareness may be unfeasible in cheap, low energy consuming networks.

ME-MAC [34] can be considered a hybrid MAC protocol because it adopts both a CAP to exchange control messages and a CFP for data transmission. As in M-MAC [35], in ME-MAC nodes are organized in clusters and cluster head (CH) election is the same in both cases [68]. Moreover, like in M-MAC, nodes are aware of their position and estimate their mobility state using an AR-1 model [67]. In ME-MAC cluster heads assume a leading role and are the destination of all packets of cluster members. Time is divided into frames and each frame starts with a SYNC message broadcast by the cluster head ; during SYNC transmission, all members must be in listening mode and synchronize their schedule with CH. After SYNC packet reception, there is a contention period during which each node that has some data to send and all nodes that are going to leave or join the cluster must send a short advertisement packet to the CH. In this way, the CH is aware of the current cluster state and can broadcast an updated schedule message. Data transmission occurs during the following period and is completely contention free.

O-MAC (Off-MAC) [36] protocol aims at limiting power consumption of the network by increasing the efficiency of the receivers. In O-MAC, time is organized in slots ; each node chooses a slot during which it will be active and ready to receive messages similarly to Crankshaft protocol [23] discussed below. Then, nodes exchange their slot so that each node can maintain a list of its neighbors addresses and the slot that they control. When a node wants to send a message, it buffers it until the intended receiver wakes up. When the receiver wakes up, it starts sampling the channel, if nothing is sensed, it turns its radio to sleep mode. Otherwise, it remains awake until the transmission ends, then it sends an ACK message to the receiver.

2.2.2 Synchronous Duty Cycle Protocols

The protocols of this family aim at coordinating node wake-ups and sleep periods to achieve a synchronous duty cycle. Communication is concentrated during active periods so that energy waste due to *idle listening* is limited. Nodes alternate simultaneous active and inactive periods ; therefore, this approach requires the node to be synchronized at network level.

S-MAC [10] is the first MAC protocol for WSNs that introduced the idea of periodic synchronous active and inactive periods. S-MAC, whose operation is depicted in Fig. 2.3, uses active periods to run a CSMA/CA MAC protocol based on the IEEE 802.11 DCF [66]. The network is divided in virtual clusters (VC) in which nodes have the same wake-up/sleep schedule. If two nodes have different schedules, that is the case of nodes belonging to two different VCs, they are unable to communicate. To maintain a fully connected network, border nodes are used to connect different VCs. To connect two VCs, border nodes must follow both schedules. Nodes of the same VC need to be synchronized to avoid time drift between each other. Synchronization procedure requires long periods in which all nodes exchange their schedule, then it is periodically updated using SYNC packets sent during the active period. When multiple nodes want to transmit to the same destination, they wait for the node to wake up and run a CSMA/CA procedure with Request-To-Send (RTS), Clear-To-Send (CTS), data and acknowledgment (ACK) messages exchange. When a long message has to be sent, it can be fragmented and sent as a burst, which avoids to repeat the RTS/CTS handshake per each packet and waste energy.

Periodic and synchronous sleep/wake-up schedules result in *sleep delays* due to the need for nodes to wait for the next active period to send data. S-MAC with Adaptive Listening [49] copes with this problem : if nodes overhear neighbor messages, they can learn the instant at which they will end their transmission and clear the channel ; then, they all turn their radio to sleep mode and wake up when channel will be clear (even though this happens during a sleep period) to exchange messages (cf. Fig. 2.3).

The time-out MAC (T-MAC) [11] protocol is an evolution of S-MAC in the sense that it proposes an adaptive duty cycle. T-MAC adopts the same synchronization method of S-MAC and nodes are organized in virtual clusters as well. The main difference with respect to S-MAC is that the active period has variable duration depending on the real needs of the node. Using a variable active period permits to reduce *idle listening*, because if no communication is going on, nodes can immediately turn the radio to sleep mode, thus reducing its active period (cf. Fig. 2.3). The amount of energy savings depends on

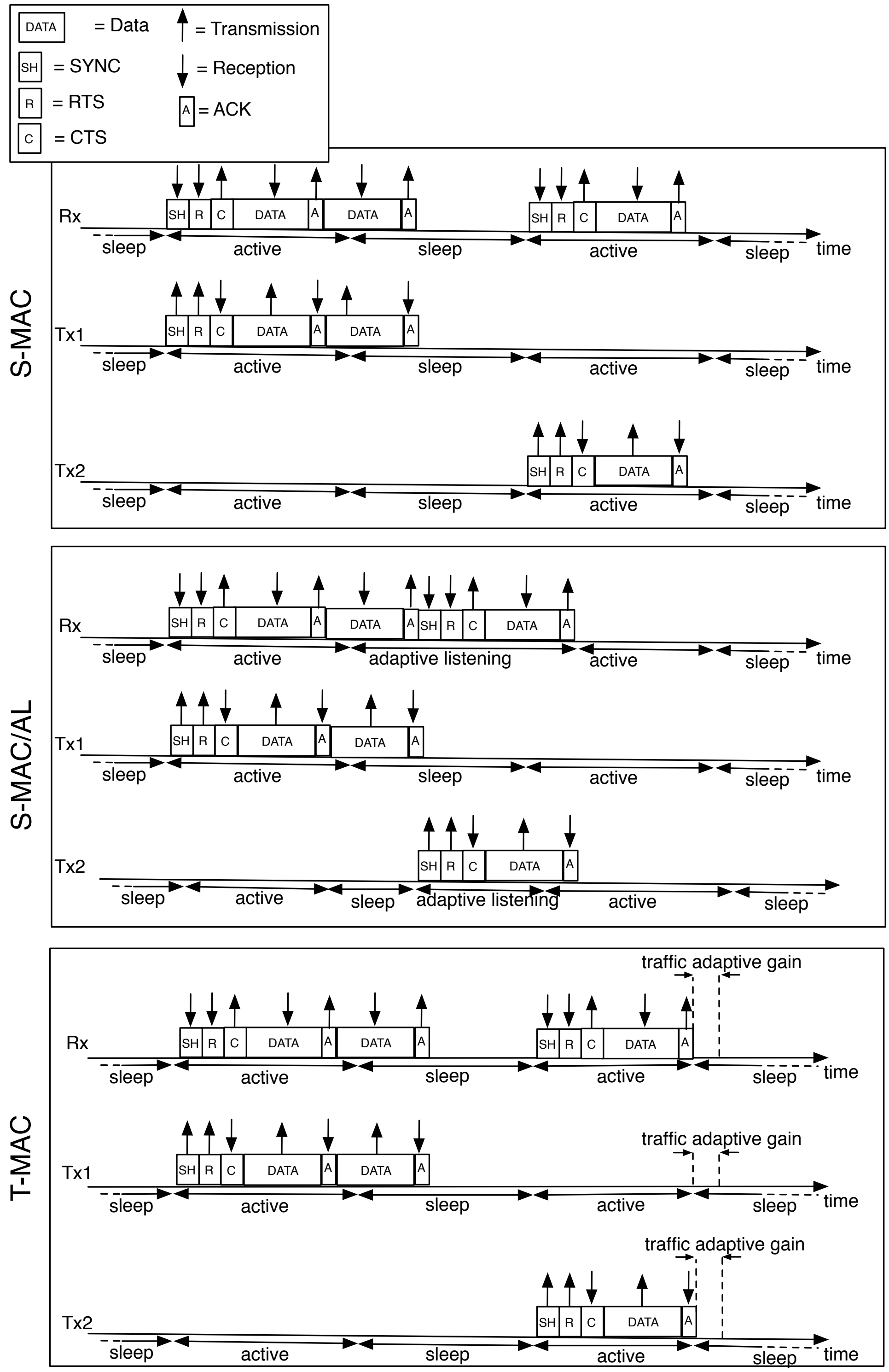


FIGURE 2.3 – Duty cycle comparison of S-MAC, S-MAC with Adaptive Listening and T-MAC.

the amount of time removed from the initial active period. As S-MAC, T-MAC must face the problem of *early sleeping* that causes *sleep delays*. To solve this problem, nodes use Future-RTS frames to inform nodes that are two hops away that a message will be sent very soon so that they wait before going back to sleep. Along with energy saving, other consequences of adaptive active period and FRTS messages are increased latency and decreased throughput. To guarantee that nodes does not miss useful communications after the SYNC period they have to wait a timeout that should be appropriately set. If during the timeout nothing is sensed, the radio can be turned off until the next frame.

MS-MAC [46] is an evolution of S-MAC protocol [10] supporting mobile environments. In S-MAC, all members of a virtual cluster periodically perform a *neighbor discovery procedure* to discover new nodes. Neighbor discovery consists in keeping their radio in listening mode for a duration of around 10 seconds to detect possible messages of any unknown source. The innovation of MS-MAC consist in adapting the period of *neighbor discovery procedure* and making it dependent on the mobility degree of nodes.

In MS-MAC, nodes are not supposed to know their position, thus, they do not estimate their mobility ; nevertheless, thanks to the variation of received signal strength of SYNC packets, they are able to *detect* if a neighbor is moving or not.

When a node detects mobility of a neighbor, it also registers the variation of the received signal strength so that its speed can be evaluated. In the case of multiple moving neighbors, the period of discovery procedure is adapted with respect to the fastest node. When a node adapts its schedule to mobility of neighbors, it adds mobility information inside its next SYNC packets (max estimated speed among all neighbors) so that all neighbors can update their periodicity and keep the cluster synchronized.

In MS-MAC, when one node moves, its neighbors create an active area around it by adapting their wake-up schedule as a consequence. The advantage of this procedure is that when a node moves from a virtual cluster to another one, the time spent while disconnected from the network is limited. As a counterpart, the short time that each mobile node spends disconnected is paid in terms of consumed energy of its neighbors that must remain awake for a longer time.

In S-MAC with Extended Kalman Filter (EKF) [47] protocol, nodes of each transmitter-receiver couple are capable of computing their relative speed and the effect of the Doppler shift (absolute position and speed are estimated). So they can predict the optimal frame size that keeps Packet Error Rate (PER) under a given threshold and adapt it as a consequence. The optimal frame size is inversely proportional to the relative speed of the transmitter-receiver couple.

Data-gathering MAC (D-MAC) [8] works for WSNs supporting convergecast communications ; that is, with D-MAC all traffic generated by nodes must be unidirectional and must converge to a unique sink node. The resulting network topology is a tree rooted at the sink. In D-MAC, wake-up instants of nodes are shifted with respect to the others depending on the position in the routing tree. In particular, each node in the tree has its active period that partially overlaps the active period of its next-hop and partially the active period of its previous node. In this way, a message can be forwarded from the source to the sink without any delay. Although efficient with converge traffic directed to one sink, D-MAC may suffer from network dynamicity such as mobility of nodes and different traffic patterns.

The routing-MAC (R-MAC) [48] protocol is a synchronous duty cycle MAC scheme that exploits routing information to reduce latency in multi-hop sensor networks. Nodes are assumed synchronized and each cycle is divided in three periods similarly to S-MAC :

synchronization period, *data period* and *sleep period*.

The period that differs from S-MAC is the data period : when a node wants to send data, it previously sends a PION (pioneer frame) message to its intended receiver to announce its data transmission. If the receiver is not the final destination of the packet, this last node immediately sends another PION to its next-hop and so on until a prefixed number of hops (similarly to FRTS used in T-MAC). This way each packet can be forwarded multiple times during the same operational cycle.

In a similar way to R-MAC, the Demand Wake-up (DW-MAC) [22] protocol is a synchronous duty cycle MAC that aims at solving the problem of packet forwarding in multi-hop networks. Like in R-MAC, when a message needs to be sent, the sender wants to alert several nodes in the path toward the final destination to prevent them from going to sleep too early. In DW-MAC, medium access control and scheduling are coupled. Sender and receiver in fact, are able to agree to the time instant at which data packet must be sent just by sending/receiving the control message, called SCHEDULE message. As S-MAC and partially also T-MAC, and R-MAC, DW-MAC suffers from the forwarding interruption problem that is solved in D-MAC with perfect synchronization of nodes.

2.2.3 Random Access Protocols

Random access protocols easy apply to WSNs and energy efficient because of the absence of signaling messages for channel access synchronization. The lack of synchronization results indeed in limited protocol overhead ; however, it may severely expose nodes to problems such as *idle listening*, *overhearing*, and collisions.

Several groups of MAC belong to the random access class, few of them are represented by a single protocol but others have been the subject of a large research activity during last years. Among these, we can cite the *Preamble Sampling* (PS) group, *Carrier Sense Multiple Access* (CSMA) group and the *multiple radios* group.

Basics on CSMA Based Protocols

A way of reducing the probability of collisions to occur is using contention windows (CW). With a contention window, two or more nodes contend for channel access in the following way : each one chooses a random number that corresponds to a random slot between 0 to *K-1*, with K being the size of the contention window. Nodes use the chosen slot to initialize a counter starting from to zero to the chosen slot. During the waiting time, each node senses the medium for channel activity. When the counter of the node with the smallest slot fires, if no channel activity was detected so far, the node asserts that channel is clear and can start its transmission. Other nodes will perceive busy channel as soon as the first node starts transmitting its frame and leave the contention. With a contention window, a collision may occur if two or more nodes select the same slot to initialize their counter.

Contention windows coupled with efficient channel sensing are the basis of CSMA protocols used in different wireless systems such as IEEE 802.11 [66]. We now briefly describe a few protocols that are no specifically designed for WSNs but have inspired the research on MAC for WSNs.

Some protocols in the literature use variable CWs whose size is updated each time a collision or a successful transmission occur. CW size adaptation has the goal of limiting packet collision and limiting idle listening. In general, when stations detect a collision they react by increasing CW size ; then, as far as a packet is transmitted with success the CW size is simply reduced to the default value (IEEE 802.11 [66]) or processed following some rule (Idle Sense [69]).

According to the 802.11 Distributed Coordination Function (DCF), each station must contend with neighbors within a CW that has not a fixed size. When a station needs to transmit, it chooses a slot $k \in [0, CW]$ to set a counter and listen to the channel (radio equipment is in channel sensing mode). If $k = 0$, the transmission begins immediately. Otherwise, if any radio signal is detected during the first time slot, the counter is decreased by one step. When the counter reaches the selected slot and any communication is overheard transmission can start. At the end of a transmission, CW size is updated to CW_{min} if the transmission was successful, and to $min(CW_{max}, 2 \cdot CW)$ if a collision occurred. This procedure is called Binary Exponential Back-off (BEB). Typical values are $CW_{min} = 32$ and $CW_{max} = 1024$. The station that wins the contention transmits a Request To Send message to the destination. If there is no collision and the receiver can decode the message, it will answer with a Clear To Send message that enables the sender to transmit data. Correct data reception is acknowledged with ACK messages.

An efficient evolution of 802.11 is Idle Sense [69]. After each transmission, successful or not, the node keeps track of the number of time slots that it has spent listening to the channel before transmission. After 5 transmissions, the node computes the average waited time slots. If this number is inferior to threshold value 5.68, the CW size is updated by $CW = min(CW_{max}, CW \cdot 1.2)$. Otherwise, the new CW is given by : $CW = max(CW_{min}, 2 \cdot \frac{CW}{2 + 1e - 3 * CW})$.

With CONTI [70] stations periodically contend channel access within a constant size CW called Contention Repetition Period (CRP). CRP is composed of a given number of time slots. Nodes that need to transmit wake up at the beginning of CRP and during the first time slot do :

- transmit a radio signal with probability P
- listen to the radio signal transmitted by others with probability $Q = 1 - P$

If a station that has decided to listen to the channel detected a radio signal, it claims itself out of the contention and switches its radio off. Otherwise it repeats the procedure. In this way, at the beginning of the second time slot only some nodes *survive* to the first slot contention and participate in the contention during the second slot. Probabilities P and Q are differentiated per time slots and are constant. The station that remains alive and alone will win the channel access. The drawback of CONTI it is that the probability of transmitting and listening at a given time slot are fixed before network deployment.

To improve CONTI approach, is possible to periodically update probabilities [71]. In this case, the values of probabilities P and Q is updated at each time slot and take into account the fact that the node has emitted or not a signal in the previous rounds.

CSMA Based Protocols for WSNs

In CSMA with Adaptive Rate Control (CSMA/ARC) [50], nodes use an energy saving version of CSMA that avoids RTS/CTS exchange. When a node needs to send a message, it previously senses the channel for a random time that depends on its application layer packet generation period ; this results in low back-off time and low collisions. In CSMA/ARC, nodes aim at ensuring fairness in multi-hops networks, to do so they give a fair proportion to the number of messages that a node generates and the number of messages generated by other nodes that must be forwarded. With CSMA/ARC nodes use implicit ACK to save energy and overhead : if a node wants to know if a message is delivered to its next-hop, it must wait until the next-hop try to forward the message to its own next-hop.

Sift [61] is a random based medium access control based on non-persistent CSMA similar to the IEEE 802.11 access protocol. The goal of Sift is limiting average latency thanks to low collision probability. In Sift, nodes choose one transmission slot number within CW according to a geometrically-increasing probability distribution instead of the common uniform distribution used in IEEE 802.11. Moreover, contention window duration is never increased even if some collisions are detected (in IEEE 802.11 if collisions are detected the CW size is increased according to the BEB rule). When a node wants to transmit, it chooses a time slot number k within the range [1,CW] according to an approximation of a geometrically-increasing probability distribution that depends upon CW and a parameter α that is constant and must be chosen off-line. The use of a non-uniform probability distribution results in low collision and permits to maintain the CW small throughout the entire network lifetime independently of collision occurrence.

BuzzBuzz [54] analyzes the problem of *Cross Technology Interference* (CTI) between IEEE 802.11 and IEEE 802.15.4 systems that both use the 2.4 GHz ISM band. Two regions of interference are identified, a *symmetric region*, in which IEEE 802.11 corrupts the reception of IEEE 802.15.4 packet header and an *asymmetric region*, in which IEEE 802.11 signal is so strong to corrupt the entire packet reception of IEEE 802.15.4. BuzzBuzz modifies IEEE 802.15.4 MAC, so when a packet must be sent, the sender first attempts to send it using an ARQ with three re-transmissions. If the packet is not delivered, the node assumes to be in the asymmetric interference region and attempts to send other three times the same message adding one extra header in front of the actual one and a Forward Error Correcting (FEC) code.

Preamble Sampling Protocols

The major goal of WSNs applications is to save energy (aimed at increasing network lifetime). The preamble sampling technique represents an extremely interesting approach because of the very limited cost due to overhead, and because of the fair balancing of energy consumption between simple sensors and sinks. All inactive receivers periodically alternate long sleeping periods (same duration for all nodes) and short listening periods (same duration for all nodes). According to the preamble sampling indeed, receivers are allowed to periodically switch to idle mode and wake-up for sensing the channel. If a node needs to send packets it precedes its data transmission with specific *preamble* packets that have the unique goal of forcing the intended receiver to remain awake for data reception. To be sure to wake up a neighbor node, the transmitter must send a preamble whose duration must be long enough to cover the sleep period of the receiver. The preamble sampling approach was coupled with Aloha [12] and CSMA [72].

With B-MAC [13], all devices periodically repeat the same cycle during their lifetime : wake up, senses the channel, and then go to sleep. Operation of B-MAC is shown in Fig. 2.4.

Node wake-up schedules are not synchronized and uncoordinated, *i.e.*, nodes are not aware of the wake-up schedules of their neighbors. All nodes belonging to the same network use the same wake-up interval. To *wake up* the intended receiver, that is, to advertise to the intended receiver the interest for transmitting a data frame, each transmitter sends a long advertisement message called preamble message. Length of preamble must be at least equal to wake-up interval.

When a node wakes up, it switches its radio to idle mode and polls the channel in order to detect any radio activity in its vicinity. Channel activity detection is performed with

Clear Channel Assessment (CCA) technique consisting in sampling the level of measured received power and compare the samples with a noise floor, if the level of several consecutive samples is lower than the noise floor, the channel is claimed clear, busy otherwise. If the channel is busy, the radio switches to receiving mode until the complete reception of both preamble and data frames.

After the end of the polling period, if the channel is clear and if the node has no data message waiting in its buffer, it switches the radio to sleep mode for a long period.

If a node wants to transmit a data frame, it needs *to wake up the receiver* ; to do so it transmits a preamble of duration at least equal to the duration of sleep period of the receiver. After preamble transmission, the sender immediately transmits its buffered data frame.

With B-MAC, devices do not need synchronization, as a consequence it results in low overhead ; nevertheless, energy efficiency is also low because to receive a data frame, a parent node must also receive a portion of a long preamble. Moreover, the use long preamble results in latency increment especially in multi-hop networks where data frames must be relayed to reach their final destination.

Preamble messages are broadcast messages, hence, if a node that is in the vicinity of the sender receives a preamble it cannot know if it is the intended destination until both the preamble and data messages are received. Therefore, the use of long broadcast preamble results in very high *overhearing* costs, especially in dense networks where the number of neighbors is high.

The B-MAC+ [51] protocol is an evolution of B-MAC that results in some energy savings. In B-MAC+, long preamble of B-MAC is divided into consecutive "chunks" that embed a "countdown" indicating the end of preamble transmission. When the receiver wakes up and detects the countdown packet, it goes back to sleep and it will be ready in receiving mode at the precise time at which data packet will be sent. Moreover, along with the countdown is also specified the address of the receiver so that nodes that are not involved in the communication will remain asleep until their next wake-up period. B-MAC+ introduces a very simple modification with respect to B-MAC but it permits to save precious energy. However, B-MAC+ still shows high latency.

CSMA-MPS (CSMA with Minimum Preamble Sampling) [14] and similarly X-MAC [15], seek to solve the overhearing problem caused by long preambles of B-MAC. The protocols are similar, therefore, in the following of this dissertation we present in details one of them : X-MAC. As in B-MAC, nodes periodically repeat the same cycle during their entire lifetime : wake up, listen for the channel, and then go to sleep (cf. Fig. 2.4). The difference with B-MAC is that if a node has buffered data to send, it transmits a series of short unicast preambles spaced with gaps instead of a unique long preamble. During a gap, the transmitter switches the radio to idle mode and expects to receive an ACK from the receiver. When a neighbor node wakes up and receives a preamble of whom is destination, it sends an ACK back to the transmitter to stop the series of preambles, which reduces the energy spent by both devices. After the reception of an ACK, the transmitter sends a data frame as shown in Fig. 2.5. Differently from B-MAC, short preambles of X-MAC are unicast,—they are exploited to limit overhearing effect of neighbor devices.

To increase network capacity with respect to B-MAC, when a receiver ends data frame reception, it waits for a possible extra data frame to come instead of immediately going back to sleep mode. To send data as a *second transmitter*, that is, during the *extra back-off time*, a transmitter must overhear a preamble destined to the same next-hop it would like to send to, if this happens, it stops sending preambles and turns its radio to idle mode until an ACK is overheard. Then, it generates a random time value t_b to initialize a back-off timer at which it will directly send data (without any preamble). Back-off timer duration is long enough so that the first transmitter has the time to complete its transmission. When the timer fires, the node performs CCA and if channel is clear, it sends extra data. Overhead is low (since devices are not synchronized), moreover strobed preambles limit the energy waste due to *overhearing effect* with respect to B-MAC. Similarly to X-MAC and CSMA-MPS, Multimode-Hybrid MAC (MH-MAC) [52] use series of preambles with gaps instead of long preambles as in B-MAC.

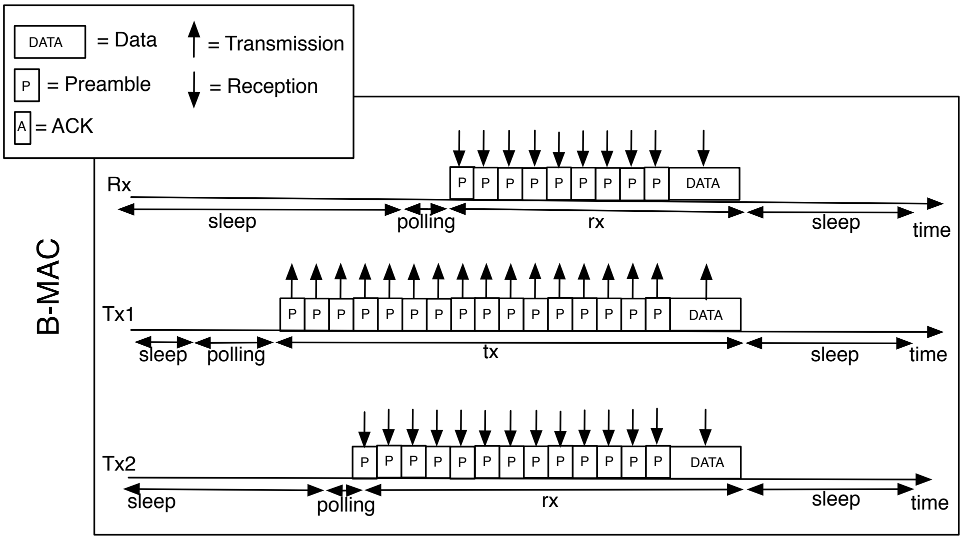


FIGURE 2.4 – Operation comparison of B-MAC, B-MAC+and MFP.

In SpeckMAC-D [62] and similarly in MX-MAC [25], if a sender wants to transmit a packet to a receiver, it performs a CCA procedure, and if channel is clear, it starts repeating the message packet for at least a given duration. When a receiver wakes up, it checks the medium. If busy, it listens until it has received a full data packet or until it realizes that it is not the intended destination for the packet.

Machiavel [59] adds mobility support to B-MAC protocol. In Machiavel, mobile nodes have higher transmission priority with respect to static nodes. When a static node ends the transmission of a long preamble, it waits a Machiavel Intern Frame Space, (MIFS) before sending actual data. If a mobile node overhears a preamble from a static node, it waits until the end of a long preamble and then it sends data during MIFS period. In Machiavel mobile nodes must be conscious that they are mobile.

Wireless Sensor MAC (WiseMAC) [16] protocol proposes a medium access scheme based on preamble sampling like B-MAC with the improvement that overhearing is reduced thanks to the knowledge at the transmitter side of the duty cycle schedule of neighbors. Unlike B-MAC, according to Wise-MAC when a node needs to transmit a data, it is aware of the exact instant at which the intended receiver will wake-up (cf. Fig. 2.5). Thus, instead of transmitting a long preamble, it can start sending a very short preamble just before the wake-up instant of the receiver. If the information about the next wake-up instant is not available at the sender side, it uses a long preamble as in B-MAC.

Adopting the Wise-MAC technique results in energy savings both at the sender and at the receiver side, because the first transmits only when necessary and the latter does not need to remain awake for the entire duration of the (long) preamble before the reception of actual data.

In Wise-MAC, each node embeds its next wake-up instant inside each ACK message that follow data transmission ; in this way, all nodes can learn the instant of wake-up of their neighbors with a passive approach. Each time that a node receives an ACK frame, it updates its local table with all wake-up instants of neighbors to compensate clock drift.

The Micro Frame Preamble MAC (MFP) [55] protocol is a preamble sampling protocol that aims at reducing energy consumption both at the sender and receiver side. Instead of long preambles such as in B-MAC, the sender transmits short messages called *micro frames* separated by a gap (see Fig. 2.4). Each micro frame embeds details about the data frame content such as the destination of data message, the instant at which last micro frame will be sent and a digest of the data field. When the receiver wakes up and receive a micro frame destined to it, its knows when data transmission will occur so that it can immediately go back to sleep, similarly to B-MAC+. In MFP, when a receiver wakes up and receive a micro frame, it also can decide whether it is interested in receiving the announced data or not, for example if the announced data is a duplication of a broadcast message already received, the receiver will goes back to sleep until the next wake-up interval.

In BOX-MAC [60], nodes adapt the preamble duration and the wake-up period depending on traffic load. By default, all nodes use strobed preamble sampling as in X-MAC and CSMA-MPS with the wake-up interval equal to preamble duration. After the reception of a message, the receiver assumes that another message may come very soon and it cuts back its wake-up interval by a prefixed amount of time, that is, it will spend less time in sleeping mode. If the sender needs to send another message to the same receiver, is now allowed to use a shorter preamble series to be sure to wake-up the receiver, thus saving energy and time. If the sender has another packet to send to the same receiver, it can use the short preamble. With this procedure each sender stores the preamble duration that must be used to transmit to each neighbor. If any message is received after a time-out, each receiver switches back to the default wake-up interval.

With the Receiver Initiated-MAC (RI-MAC) [57] protocol, the communication is initiated by the receiver. Nodes periodically wake-up at random time and send a beacon to advertise neighbors that they are ready to receive data. If a neighbor node is awake and has waiting data to send, it waits a random back-off time to limit the probability of collisions, senses the channel (CCA) and if it is clear, transmits data. Immediately after packet reception, the receiver sends another beacon to acknowledge the correct packet reception. The transmission of the second beacon has also the goal of announcing that the receiver is ready for additional data transmission.

Another MAC (A-MAC) [58] improves performance of RI-MAC by introducing *auto-ACKs* in response to receiver beacons. After auto-ACK transmission, data are sent using a contention window to reduce the collision probability. If multiple senders have messages to send, auto-ACK messages may collide ; in A-MAC, even if auto-ACKs collide the receiver is able to decode the information and decide to remain awake to receive data. If data messages collide, the sender increments the size of the contention window that will use when the next beacon will be received. The operation comparison between RI-MAC and A-MAC is shown in Fig. 2.6.

Multiple Radio and Multi Channels Protocols

In Framelet-MAC (f-MAC) [63], each message to send is decomposed in a number of

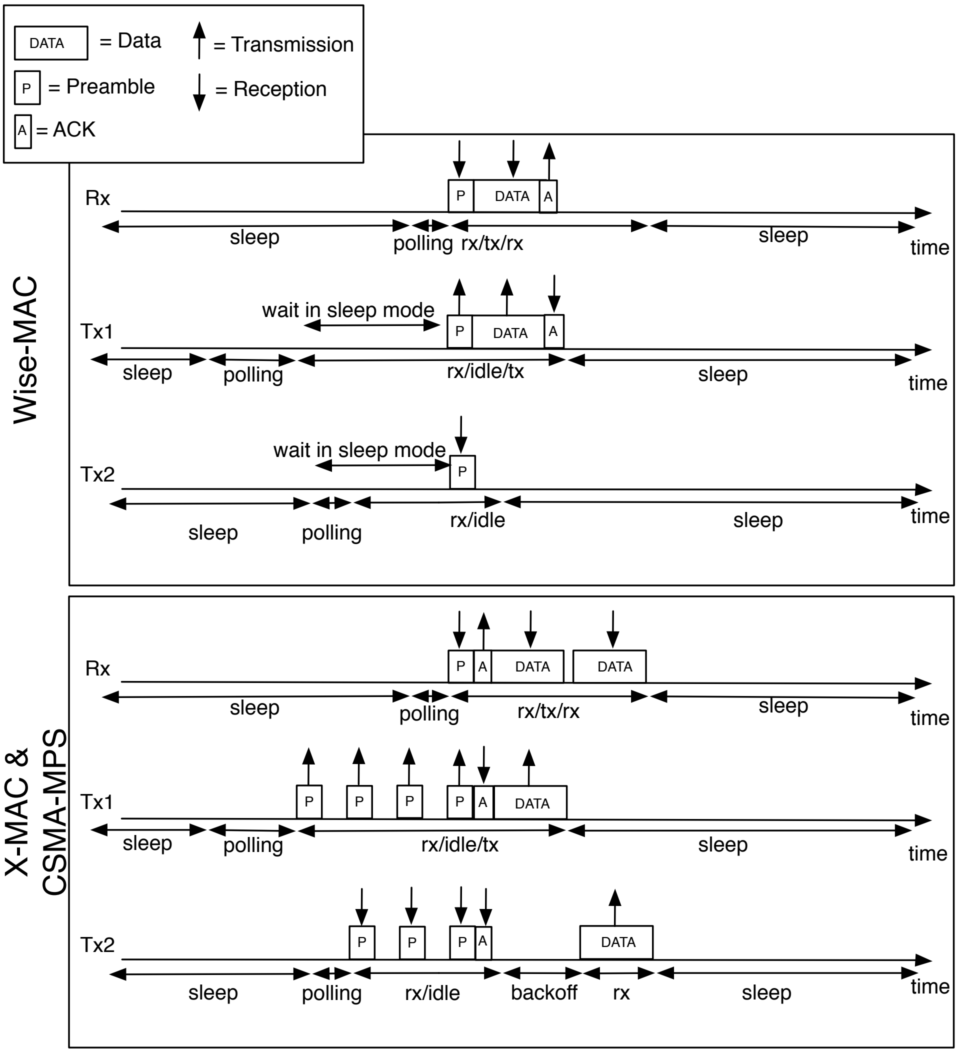


FIGURE 2.5 – Operation comparison of WiseMAC and X-MAC.

constant size small packets (*framelets*) that are sent at a given frequency. Different nodes operate at different frequencies, thus, collisions are limited and wake-up synchronization is unnecessary. In particular, with f-MAC nodes choose the number of framelets per message and the transmission frequencies in a way that guarantees that at least one framelet per message is delivered without collision. The advantage of f-MAC is that the message delay is guaranteed by un upper-bound, however, energy savings are limited because each node must periodically check channel activity and receive framelets.

In PicoRadio [53], nodes use two separate radios : the first, which is a very low power single channel radio, for wake-up tone transmission and the second, which is a multi-channel CDMA radio, for data transmission. The protocol is organized in two phases : a setup phase for orthogonal CDMA code selection and the actual data transmission. During the first phase, nodes randomly choose a dedicated channel among a pool of available channels and then use a common control channel to exchange their selection. If a node detects a channel collision, it chooses another channel and it broadcasts again its choice on the common

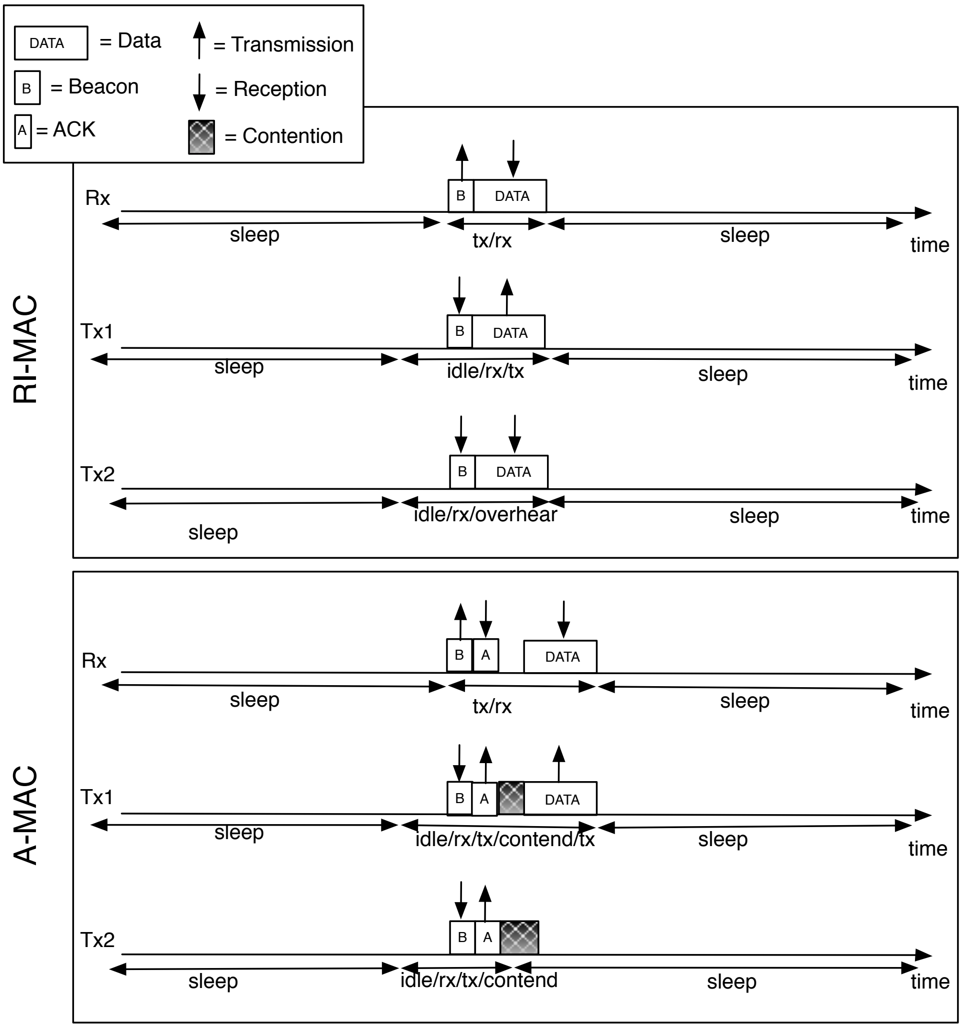


Figure 2.6 – Operation comparison of RI-MAC and A-MAC.

channel. In PicoRadio, static nodes quickly converge to a collision free channel assignment. At the end of the setup phase, all nodes are aware of the channel that is controlled by all neighbors. When the setup phase is ended, all nodes turn off the multi-channel radio and sense the channel using the low-power radio for a tone to come. In PicoRadio, tones are unicast, that is, they can carry the address of the destination. When a node receives a tone destined to it, it turns its multi-channel radio on, it tunes it to the channel corresponding to the sender and receives the data message. PicoRadio can support mobile nodes by changing the channel allocation rule and if mobile nodes are aware of their own mobility. The major issues of this protocol arise from the sensitivity of low-power radio that may result in false positives.

The sparse Topology and Energy Management (STEM) [24] protocol is a topology management scheme that seeks to increase energy savings in WSN communications by the adoption of the preamble sampling technique and the use of dual radios : one for signaling and the other for data transmission.

In STEM, nodes that want to initiate communication are called *initiator* nodes while nodes that will receive packets are called *target* nodes. In STEM, nodes do not need to be synchronized thanks to the preamble sampling procedure, thus, overhead is limited. STEM consists of two preamble variants : the first one that uses a series of beacons with gaps (STEM-B) containing the addresses of the target node, and the second one that uses tones (STEM-T). Both variants of STEM send control messages on a dedicated channel to wake up the receiver. STEM-T is a multi-channel precursor of B-MAC. In STEM-B, when no packets are to be sent, nodes remain in sleep mode and periodically sample the paging channel using a low power radio. If a node detects a beacon announcing a message for it, it sends an ACK to the initiator using the paging channel, then it switches its primary radio on to receive actual data. When the initiator receives the ACK, it also switches its primary radio on and send data.

RATE EST (Rate Estimation MAC) [56] uses multiple radios to define a protocol similar to WiseMAC [16].

2.2.4 Hybrid Protocols

Crankshaft [23] is a MAC scheme in which sleeping nodes wake up at fixed time slots only to receive data. If a node does not need to send anything it remains asleep all the time. The analogy of the MAC protocol with the crankshaft of an engine is that each user receives data at a fixed offset from the start of a frame exactly like the piston fires at a fixed offset from the start of the rotation of the crankshaft.

In the Crankshaft protocol time is divided into frames and each frame into several slots. Two types of slots are possible, unicast slots and broadcast slots. When a node wants to send a message to a destination node, it must wait for the destination receiving slot to come and just before the beginning of the slot it contends the channel access with other possible senders using the Sift technique [61]. In Crankshaft, data are acknowledged. Special nodes like the base stations and the sink nodes do not follow the same wake-up/sleeping schedule of normal nodes because they must be always awaken. Crankshaft avoids collisions and *idle listening* during unicast slots, nevertheless, all nodes must be in idle mode during broadcast slots so that energy is wasted if no packet is transmitted. Crankshaft requires perfect synchronization of nodes, therefore, each transmitted packet embeds information about synchronization in the header.

Scheduled Channel Polling (SCP-MAC) [17] is an evolution of S-MAC that integrate preamble sampling technique and two constant size contention windows to avoid collisions and *idle listening*. In SCP, all devices are synchronized to the same schedule, network-wide. A receiver periodically wakes up and polls the channel for any activity, if nothing is sensed it turns its radio to sleep mode. A sender uses two contention windows, the first to send an advertisement frame and the second to transmit data frames. The advertisement message is used to intercept the wake up of the receiver like in preamble sampling MAC protocols. After advertisement transmission, the sender uses the second contention window (larger than the first one) to transmit an RTS frame, waiting for a CTS and send the data frame. SCP protocol is traffic adaptive : after the transmission of first data frame, an adaptive number of additional transmission slots are used to increase the number of senders per each active period as shown in Fig. 2.7.

Zebra MAC(Z-MAC) [64] is an access protocol for WSN that exploits the advantages of both CSMA and TDMA schemes. Z-MAC is said to be “hybrid” in the sense that according to the contention level present in its neighborhood a node can decide to adopt a more structured access method (TDMA-like with one slot owned by one device) or to

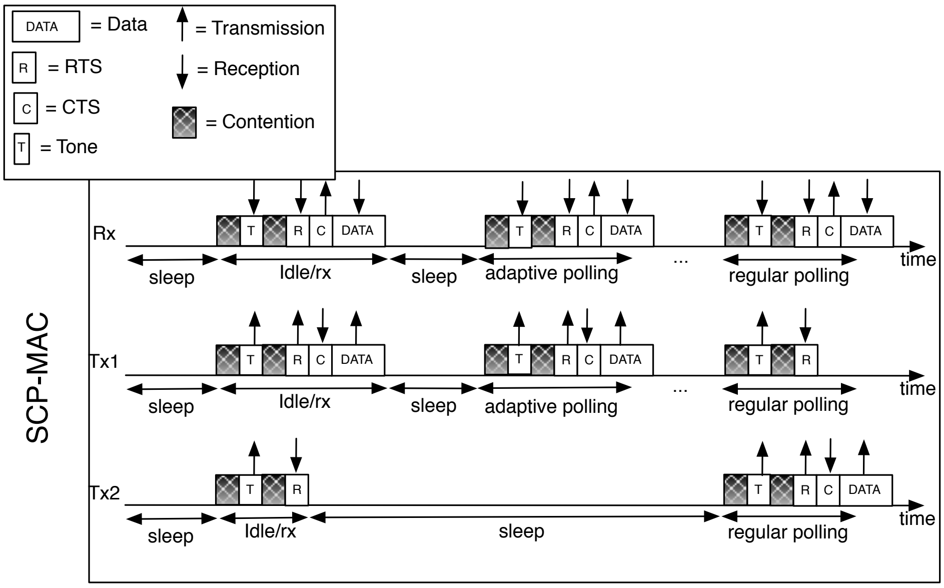


FIGURE 2.7 – Operation of SCP-MAC protocol.

contend for transmissions slots. In Z-MAC, nodes are divided in small groups and thanks to a distributed algorithm called D-RAND [73] ; each node has assigned a unique slot (slots are reused by nodes that are at least 2-hops away). If the level of contention is low, nodes are allowed to transmit even during a slot that is owned by another neighbor (if it does not need it). If the *owner* of a slot needs to transmit, it has the priority on that slot ; however, if its packet queue is empty, other nodes compete for its slot with a CSMA-like procedure. As a result, Z-MAC procedure increases throughput with variable traffic patterns.

Pattern MAC (P-MAC) [65] achieves limited energy waste thanks to adaptive duty cycle. In P-MAC duty cycle of a node depends on both its traffic and the traffic of its neighbors. When traffic load is light, a node can remain in sleep mode for long periods. In P-MAC time is divided into frames that have a constant duration and frames are divided in slots. The duration of each time slot is chosen large enough to handle a complete data transmission including contention window, RTS, CTS, data, and ACK.

In P-MAC, each node periodically generates a *pattern* in which it announces during which slots of the frame it *intends* to be active (radio turned on) and in which slots it *intends* to sleep. Then, all nodes exchange their patterns with neighbors. For each node, the goal of pattern exchange is to understand which are the *intentions* of the neighbors, compare them with its own *intentions* and determine a *schedule*. The schedule contains the information about during which slots a node will be actually active and those during which it will sleep. If a node announces in the *pattern* that it intends to be active in a given slot, it will definitely be awaken during that slot. Other nodes store in their schedules which is the slot to use to communicate to that node. Nevertheless, if any node has a message to send, the reference node wastes its energy listening for an idle channel. As a consequence, P-MAC is prone to the idle listening problem.

2.3 Handling Density Variations in WSNs

The term *density control* indicates the set of procedures aimed at maintaining the network density to a uniform level to save power. In the literature, we find several mechanisms for density control. Goal of NADC [74] is to maintain a high level of network connectivity. To achieve this goal, it exploits density information and forces some nodes to remain in sleep state for a given amount of time. In NADC, nodes periodically observe the channel to estimate the number of active surrounding nodes. If the number of active neighbors is above a given density threshold, the node decides itself whether to go to sleep state or not.

The network is logically divided in different regions or clusters depending on the fact that in a given region there is at least one source node, only relaying nodes or if the density of nodes is very low. Nodes belonging to different regions use different density thresholds.

AFECA protocol [75] is based on the idea that in densely-populated networks, nodes that are equivalent from a routing perspective, are interchangeable. The strength of the protocol comes from the estimation of local density to increase the sleep time of nodes, if density is high enough to guarantee reliable routes to the destination nodes.

PEAS [76] analyzes the relation between energy consumption and the distance of nodes from the data collector node (the sink) : the closer the sink, the higher energy consumption. Therefore, by increasing the density of nodes that are closer to the sink, the distribution of network energy consumption can be uniform. The PEAS protocol aims at achieving (sub-optimal) uniform energy consumption by adjusting the density of nodes and by adapting routing paths with respect to the current battery status.

2.4 Conclusions

In Wireless Sensor Networks, the major reason for energy waste are *idle listening*, that is the activity of polling the channel without the reception of any frame, *overhearing*, that is the activity of receiving a frame that is destined to someone else, and re-transmissions due to *collisions of frames* that may occur when two receptions overlap in time. Thus, while organizing channel access, MAC for WSN must be energy efficient. To save energy, devices aim at achieving low duty cycles : they alternate sleep periods and active periods. During active periods, nodes exchange frames to communicate while during sleep periods radio is switched off. The challenge of MAC is to coordinate active periods so that when there is a frame to send, transmitter and receiver wake up, communicate and go back to sleep. Coordinate active periods in large, dense and multi-hop networks is a complex problem.

Existing solutions basically adopt three techniques : schedule based protocols, synchronous duty cycle protocols, and random access protocols. We have investigated main contributions of each technique providing their strong aspects and drawbacks. We have observed how during last years some leading concepts have inspired other proposals. One of these leading concepts is *preamble sampling*. This approach is energy efficient, flexible, easy to use, and its operation outperforms the synchronous duty cycle approach [77]. As a result, more than 10 protocols present in the literature are based on this approach. The characteristics of the preamble sampling approach make it suitable for operating in dynamic networks with multiple different traffics, hence, in the following chapters we focus our attention on preamble sampling protocols (see Chapters 3 and 4). The problem of variable network density is faced with a hybrid adaptive approach based on synchronous duty cycle (see Chapter 5).

Chapter 3

Low-Latency MAC for Heterogeneous Wireless Sensor Networks : LA-MAC

This chapter introduces the problem of handling multiple heterogeneous traffic sources in multi-hop wireless sensor networks to ensure low latency. To cope with such challenging problem, in this chapter we propose a novel MAC protocol, "LA-MAC", that provides support for multiple applications, handles heterogeneous traffic, and that is suitable to work with complex topologies. LA-MAC, a low-latency asynchronous access method for efficient forwarding in WSNs, takes advantage of the structure in multi-hops networks so that a next-hop of some nodes becomes a coordinator that schedules transmissions in a localized region. To achieve this, each sender transmits preamble messages embedding its transmission requests to the next-hop. In LA-MAC, nodes are allowed to transmit bursts ; this improves the network capacity so that the network can handle load fluctuations. At the same time, the method reduces energy consumption by decreasing the overhead of node coordination. Numerical simulation reports on the results of extensive simulations that compare LA-MAC with B-MAC [13] and X-MAC [15], two energy efficient methods for WSN also based on the preamble sampling technique.

The chapter is organized as follows. After having introduced and motivated the problem in Sec. 3.1, heterogeneous sources of traffic are presented in Sec. 3.2. Sec. 3.4 presents the proposed LA-MAC method. Before drawing conclusions in Sec. 3.6, some simulation results and comparisons are shown in Sec. 3.5.

3.1 Introduction

First wireless sensor networks were typically designed for a specific application that generated one type of low intensity traffic [3, 4]. As a consequence, MAC protocols designed to support such traffics expected to deal with a communication channel that is idle most of the time. The main design goal of access methods was minimizing energy consumption, that consists of reducing the effect of *idle listening*, in which nodes consume energy waiting for an eventual transmission and *overhearing*, the reception of data frames destined to other devices [7].

Current wireless sensor network applications become multi-purpose and can convey heterogeneous traffic coming from different applications [78]. This trend benefits from the IETF ROLL (Routing Over Low power and Lossy networks) standardization effort [79] that fosters the development of more generic wireless sensor networks connected to the Internet following the way similar to the early Internet with a common communication infrastructure independent of applications.

3.1.1 Motivations

Our goal is to design an adaptive MAC protocol suitable for large-scale multi-purpose heterogeneous wireless sensor networks running a routing protocol that structures their operation. We want to take into account different types of traffic by providing support for service differentiation and support efficient multi-hop packet forwarding by making MAC to operate according to the network structure and possible multiple routes established at the network layer.

The characteristics of the new kind of wireless sensor networks, such as heterogeneity, complexity and high density require flexible MAC protocols, therefore an asynchronous approach like the *preamble sampling* approach results more appropriate than a synchronized approach. The choice of preamble sampling technique is motivated by the following reasons. First, because achieving time synchronization on a large scale and in networks with dynamic topology is difficult. Second, synchronization implies some overhead that may be cumbersome in case of traffic variations ; with the choice of the asynchronous approach, we can deal with scalability, because close cooperation of nodes required by efficient forwarding will be local to small groups of nodes. Third, because short preambles are flexible and can be exploited to convey detailed transmission requests to efficiently coordinate localized transmissions of contending nodes.

In this chapter, we compare the proposed LA-MAC protocol with two preamble sampling methods : B-MAC, the first preamble sampling MAC method to be developed and implemented and X-MAC, an energy saving version of B-MAC. Both protocols are used as reference protocols in our extensive numerical simulations (cf. Sec. 3.5) and in the energy consumption analysis presented in Chapter 4.

3.1.2 Contribution

This chapter is based on one paper accepted for publication at the IEEE Personal Indoor Mobile Radio Communications conference [C2] and a pending Patent of Commissariat à l'Energie Atomique [P1].

We propose LA-MAC, a low-latency asynchronous access method for efficient forwarding in multi-hop wireless sensor networks. The method is based on the preamble sampling approach and avoids explicit synchronization messages. Devices include the details of their channel requests in short preamble messages, then a receiver gathers multiple transmission requests and organizes local transmissions by sending a SCHEDULE message. LA-MAC operation is the following :

- Neighbor nodes are organized in a structure corresponding to the routing information (a tree, a DAG, a Clustered Tree, or a mesh). In particular, a node knows its parent (a next-hop node) on a given route.
- Potential receiver nodes (parents) periodically wake up each *wake-up interval* and wait for the reception of a series of short preambles. Nodes can adapt their check intervals to handle variations of traffic.
- Nodes contend for a transmission of a *burst* to the next-hop by sending a series of short preambles with the information that allows the next-hop to schedule transmission bursts based on priority, age of a burst, burst size. Grouping packet transmission allows to handle higher volumes of data closer to a sink and limits energy consumption by reducing the protocol overhead.
- The next-hop allocates the channel to transmitters by sending an ACK frame that defines rendezvous later on for transmitting a burst of data frames. Nodes can schedule earlier transmissions of high priority traffic.
- A transmitter node goes to sleep, wakes up at the instant of a given rendezvous, at which parent node sends a SCHEDULE message containing transmission organization.
- Transmitters follow the schedule, perform a CCA (Clear Channel Assessment), and sends their burst.

LA-MAC results in lower latency and higher delivery ratio than B-MAC and X-MAC in several multi-traffic scenarios. Moreover, it results to be more energy efficient (cf. Sec. 4).

The proposed protocol is the result of a deep investigation of PS procedure ; advantages such as energy savings and simplicity are coupled with the idea of fixing a *rendezvous* for data transmission. The idea of embedding pieces of information inside preamble messages is not new, though [15, 51, 80]. The innovation of the proposed protocol arises from combining enriched PS preambles to locally organize data transmission in a collision limited way.

3.2 Traffic Heterogeneity

Homogeneity and heterogeneity are concepts relating to the uniformity or lack thereof in a substance. A material that is homogeneous is uniform in composition or character ; one that is heterogeneous lacks uniformity in one of these qualities [81].

By analogy with a generic definition of heterogeneity, in this chapter, we deal with the issue of handling heterogeneous elements that coexist in the same WSN. In particular, we focus on heterogeneous traffic.

In a typical sensor network, wireless sensor networks have a simple tree structure reflecting the need for transporting measured data to a single sink. All data, belonging to the same class of traffic converge from leaf nodes to the root. If all data is measured on a periodic basis with constant period, resulting network traffic is predictable and severe load fluctuations are unlikely. In contrast, if measurement period differ from a node to another, packets may have different priority and some nodes may generate burst of very urgent data, network load fluctuations become very high and frequent. For example, this happens when different applications generating different classes of traffic must coexist in the same WSN. Moreover, the network may include multiple sinks and even mobile nodes or sinks. Large wireless sensor networks may be randomly deployed, thus requiring routing protocols to build routes. As an example, RPL (Routing Protocol for Low Power and Lossy Networks) [79] defines the structure of a DAG (Directed Acyclic Graph) for Multi-Point-to-Point traffic (MP2P - routing packets to a single sink) and considers multiple sinks with parallel DAGs. It begins to take into account the need for Point-to-Point traffic (P2P), which finally will result in a design of a full-fledged routing protocol between any pair of sources and destinations.

First MAC methods for WSNs were focused on energy savings in networks with homogeneous nodes with a single traffic consisting of infrequent periodic measurements. Current networks may include nodes with various characteristics (such as different computational power, type of power supply and mobility capabilities) that generate different types of traffic : sporadic alarms, periodic high-volume multimedia data (images or video), real-time control commands for actuators, etc. [82, 83]. Moreover, with networks that are large and dense, nodes close to sinks may need to transport increasingly high volumes of data with some QoS constraints, e.g. low latency and high delivery ratio.

As performance of batteries improves and alternative energy scavenging technologies appear, criticality of energy savings in the network operation becomes slightly less important. Nevertheless, optimizing energy consumption is still important, but at the same time, it is also important to take into account performance aspects such as higher throughput, lower latency, and provision for traffic differentiation. Moreover, variations in traffic call for adaptation mechanisms that adjust the operation to a given load. The last aspect concerns unicast vs. broadcast communications—convergecast traffic towards sinks is the most important, however self-organizing operation also requires some support for broadcast communications that may become a problem when neighbor nodes use different channels or wake up instants.

3.3 System Model

We start with presenting the role of nodes in a network structured according to some higher layer information (a routing tree, a DAG, or a graph). Then, we explain the rules of method operation.

We consider a traffic-heterogeneous wireless sensor network that may include several sinks and runs a routing protocol that structures its operation. Routes towards sinks form a structured topology that may go from a tree rooted at one sink to a general graph. *Simulation playground* is represented by a square plane with a surface of A m^2. The area is plane and there are no obstacles between sensor nodes. Initial node *deployment* is a problem that must be faced by all WSNs designers. There is not a standard for node deployment and each specific simulation scenario requires a given deployment model to be used. If not differently expressed, in the present work we adopt a random deployment model. Given a simulated area A and the achievable radio range of devices, random deployment model allows us to keep average number of nodes under a given threshold. Nodes are always deployed over a two dimensional plane. The number of sinks may be one or more depending on the specific scenario. If not differently expressed, nodes are assumed static.

Nodes assume some roles that can vary in time. Without a loss of generality we explain the notions below by taking an example of a tree structure rooted at a single sink.

Nodes know their next-hop nodes (parents) towards a sink and the nodes for which they act as next-hop (children). More precisely, depending on the relation with its neighbors, a node may be a *leaf node*, a *child node*, a *parent node*, a *sibling node*, an *interferer*, or a *sink*. Fig. 3.1 presents an example network composed of nine nodes and a sink with possible relationships between nodes. Node roles are not exclusive and a node that is a leaf node for a tree may be a parent with respect to another tree.

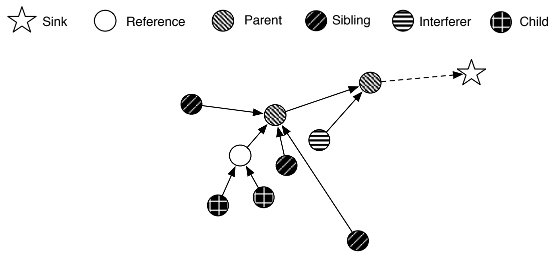


FIGURE 3.1 – Example of WSN showing the role of nodes.

In the network there can be several coexisting different traffic sources $Tr_i \forall i \in (1, M)$. Each traffic has a specific traffic priority $S_i(t)$. Source priority can vary over time according to different rules, such as constant, increasing, decreasing and random. Each node can be generator of one or more traffic sources.

We identify two major types of traffic sources, periodic measurement messages, called *monitoring* messages and sporadic *alarms*. The intensity of monitoring traffic is characterized by the Packet Generation Rate (r_m) measured in Packet Per Second (*pps*) that is the number of packets generated by a source per second. The alarm traffic has higher priority than the monitoring traffic and is characterized by the size of a burst. Alarm traffic is characterized by the "probability of generating a burst" Γ : each alarm source sets a periodic timer, when it fires, the source generates a burst of alarms with probability Γ (with probability (1-Γ) the source remains silent). Hence, an alarm source is defined by Γ, the burst size and the periodic timer value. We assume that packets of both types of traffic have the same size. In terms of timing constraints, monitoring packets do not have latency constraints, but alarm packets must reach the sink as soon as possible.

We assume that each device is provided a *single mono-channel radio equipment* and that each radio can be in one of four available modes (cf. Fig. 3.2). Antennas are omni-directional.

We distinguish the polling mode from the reception mode. When a node is performing channel polling (CCA), it listens to any channel for activity—to be detected, a radio transmission must start after the beginning of channel polling. Once a radio activity is detected, the device immediately switches its radio mode from polling to receiving. Otherwise, the device that is polling the channel cannot change its radio mode.

Each radio mode has its specific energy cost, transmission E_t, reception E_r, polling E_l, and sleeping E_s.

The power consumption of radio modes is P_t, P_r, P_l, and P_s for transmission, reception, channel polling, and sleeping, respectively. In the present work any particular radio stack is required. Simulations can last a different time depending on specific stop criterion. By default, stop criterion is based on time limit ; other possibilities are limited number of data frames to send or node battery exhaustion.

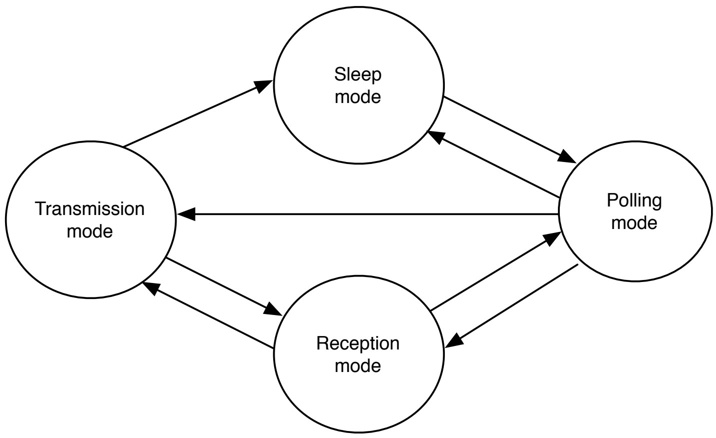


FIGURE 3.2 – Transitions among four radio modes : transmission, reception, idle and sleep.

A collision may occur if two or more frame receptions overlap for some time. The reason for frame collisions is the hidden terminal problem : assume that node A_1 is transmitting to the node A_2 and that there is a third A_3 close to A_2 but far from A_1, that wants to transmit to A_2 too. When A_3 performs CCA, it claims the channel clear because it cannot hear messages of A_1. In this situation if A_3 starts transmitting, its transmission would overlap with the ongoing one between A_1 and A_2. Unlike the unit disk model, we adopt more realistic radio propagation assumptions. All nodes transmit with the same power and their radio signal strength decreases according to the Free Space formula with $\alpha = 3$.

We adopt the *path-loss* power attenuation : the reference node can receive and decode frames with a received power larger than its sensitivity—they are sent by the nodes inside its *reception range*. Moreover, to consider multi-user interference, each node can receive (but not decode) frames whose received power is less than sensitivity, but larger than the *interference sensitivity*. The nodes that can send such frames lie inside the *interference range*. *MAC buffer* can have limited size or be unlimited, depending on the goal of the specific scenario.

3.3.1 Preamble Sampling MAC Notations

The time between two wake-up instants is called a *protocol frame time* and lasts $t_f = t_l + t_s$, where t_l and t_s are respectively the channel polling duration and the sleep period. These values are related to the duty cycle, that is, $\frac{t_l}{t_f}$. Each preamble lasts t_p^ν where ν at exponent denotes the specific protocol. It the protocol adopts ACK frames, their duration is t_a^ν. In a tree based topology, each node may be the next-hop of multiple senders, such node is called parent node and the senders are its children. The number of children of a parent i is represented by C^i. Wake-up instants are random and asynchronous, the next wake-up instant of node i is represented by $t_w p^i$.

In packetized radios, a long preamble, like in B-MAC, is obtained by a sequence of short preambles sent one right after the other [7]. For this reason, if a generic device A_2 wakes up and polls the channel while a generic device A_1 is sending a long preamble, radio mode of device A_2 remains in polling mode for a short time until the beginning of the next small packet of the long preamble ; afterwards the radio switches in receiving mode consuming more energy.

3.4 Proposed LA-MAC method

We describe in this section the principles of the proposed MAC access method. heterogeneity of sources. To simplify presentation without reducing generality, we now consider only two classes of traffic : *monitoring* traffic of almost constant low intensity and sporadic bursty *alarm* traffic with possible high variable intensity.

3.4.1 LA-MAC Protocol Description

The motivation for the design of LA-MAC protocol comes from the fact that no existing MAC access method can efficiently adapt its behavior (time of reaction, energy consumption, and latency) to the variation of some network parameters such as traffic fluctuations or node mobility to achieve low latency. LA-MAC tries to achieve this by building on three main ideas : efficient forwarding based on proper scheduling of children nodes that want to transmit, transmissions of data frame bursts, and traffic differentiation. The method periodically adapts the organization of channel access depending on network dynamics. If more nodes are active at the same time, it acts so that all of them can transmit without collisions.

In existing MAC based on preamble sampling, the cycles of node wake-up/sleep are independent and in case of N contending children nodes, sensors are scheduled in the FIFO order and they are allowed to transmit only one data frame per duty cycle. If other remaining *N-1* children nodes want to transmit, they must wait until the end of a transmission and then they can begin their own preamble sampling. In this way spectral efficiency of the system is low due to the high number of preamble packets that must be transmitted each time that a new data packet must be sent.

Note that if $N = 1$ and data traffic is sporadic, the existing preamble sampling schemes are energy efficient, conversely in the case of $N > 1$ and/or non-sporadic traffic, the data frame latency increases very much and spectral efficiency drops.

Goal of LA-MAC is to be able to support fluctuations of the network parameters minimizing protocol overhead.

LA-MAC improves spectral efficiency by letting nodes send several data frames in a burst in the same preamble sampling period. Moreover, scheduling multiple children nodes enables traffic differentiation avoiding in this way standard FIFO handling of all data frames. In addition to this, scheduling of nodes has also another advantage that arises from the exploitation of the overheard messages exchanged between a sibling node and the parent. In fact, if a node detects an occurring handshake between its parent and one sibling it can remain silent and postpone its preambles transmission right after the reception of the ACK destined to the sibling node. This way in case of multiple active neighbor nodes the amount of time spent in transmitting preambles is minimized while in existing PS protocols will increase.

Fig. 3.3 illustrates the operation of the LA-MAC protocol for monitoring unicast traffic : there are two transmitters with bursts of data frames to deliver to the same parent node. All frames to transmit have the same priority. The figure also includes operations of B-MAC and X-MAC for protocol comparison.

LA-MAC consists of several consecutive steps as follows :

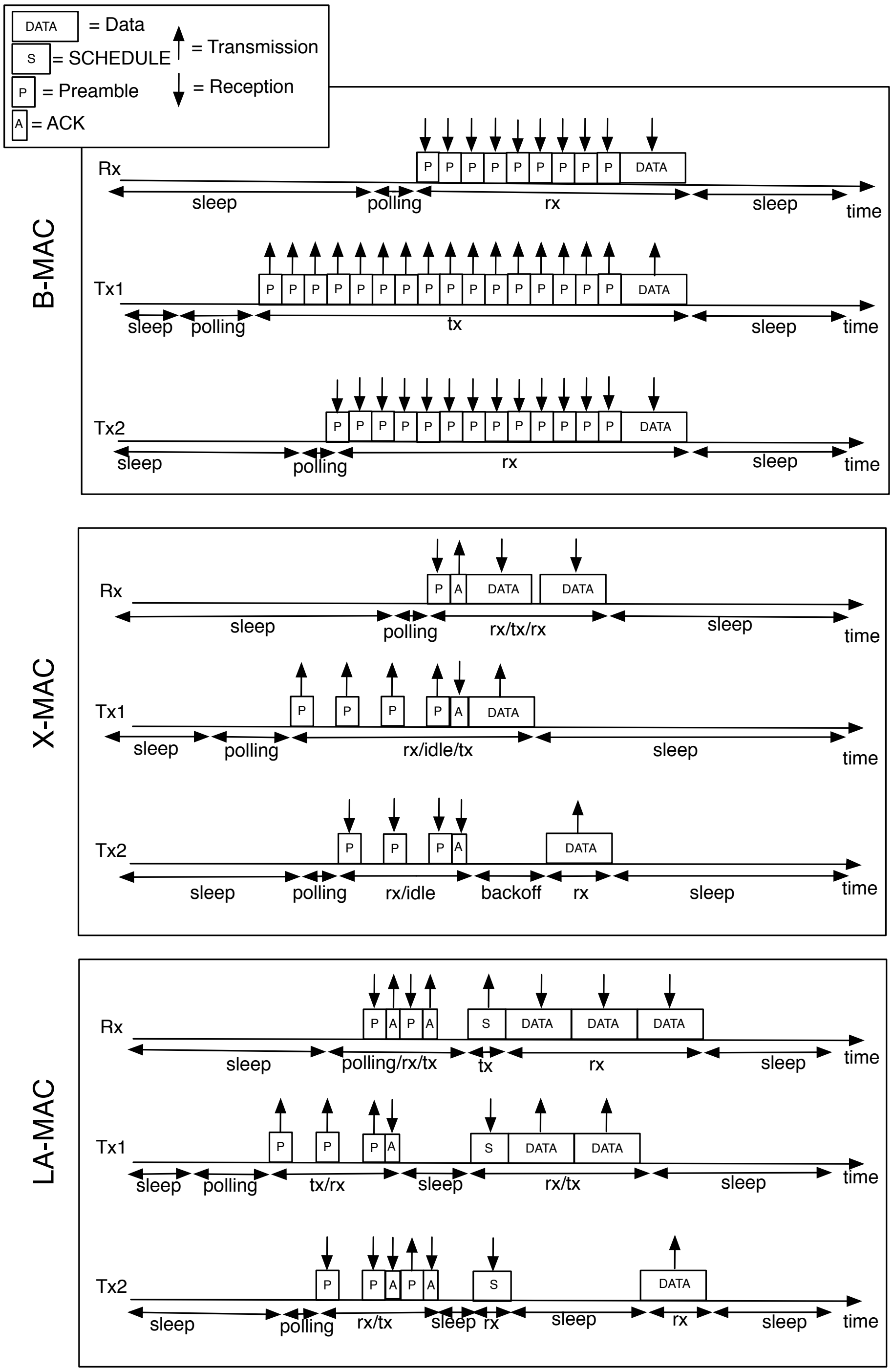


Figure 3.3 – Example of data frame transmissions with three preamble sampling MAC methods, B-MAC, X-MAC and LA-MAC.

Step 1 *: Wake up and poll the channel.*

Nodes periodically alternate long sleep periods with short channel polling periods. Channel polling duration is the same for all nodes (t_l), however nodes are not synchronized so their wake-up and polling periods may start at different instants. In Fig. 3.3, we assume that both sensor nodes have some data frames to send and that both nodes are able to decode all data frames (each node is within the transmission range of the other). Transmitter 1 (T_{x1} in the figure) is the first sensor to wake up, it turns its radio into the idle mode, and starts its channel polling (CCA) procedure. Wake-up frequency of nodes can be adapted to traffic variations ; Sec. 3.4.4 gives more details about the frequency adaptation.

Step 2 *: Preamble sampling.*
When the CCA ends and the channel is clear, T_{x1} starts sending a series of short preambles containing the information about the burst to send. As illustrated in the figure, the first preamble is not received by any of other nodes, because they are in the sleep mode. T_{x2} receives the second preamble and although it is not the intended receiver (Rx in the figure) of this preamble frame, it interprets the preamble as a blocking signal so it remains silent until the reception of the parent ACK, it then can start transmitting its own preambles.

A short preamble conveys some information about the status and the size of its local packet queue :

- *Destination*, the address of the parent node,
- *Priority of the burst*, the highest priority of data frames in the burst,
- *Age of the burst*, the age of the highest priority data frame in the burst,
- *Burst size.*

Thanks to preamble overhearing neighbor nodes can act to reduce reciprocal interference. If a node overhears a preamble that is neither destined to itself nor to one of its next-hop nodes, it turns its radio to sleep mode to not disturb ongoing communications. Otherwise, if the destination of the overheard preamble is one of the next-hop nodes, the overhearing node takes advantage of the fact that another node is trying to wake up one of its parents and waits for that parent ACK in idle mode. When the ACK is received, overhearing node can send a preamble frame. Despite the advantage of the overhearing effect, it may happen that a node overhears a preamble but not the relative ACK ; in order to avoid deadlock states, nodes use a time-out whose maximal duration is equal to the size of the channel polling period, then they go to sleep for a duration of an entire protocol frame (t_f). Similarly to X-MAC, with LA-MAC the average number of preambles needed to *wake up* the receiver is given by γ^L :

$$\gamma^L = \left(\frac{t_l - t_a^L - t_p^L}{t_f} \right)^{-1}, \tag{3.1}$$

Step 3 *: Preamble clearing with a rendezvous.*
When the parent node receives and correctly decode a preamble (no collision occurs), it immediately "clears" it by sending back an ACK frame containing the instant of a *rendezvous* at which the parent will broadcast a SCHEDULE frame, that is a message containing the result of its scheduling decision. An ACK also forces a sender to stop transmitting short preambles so that another child can transmit its preamble with success. After receiving an ACK, a child goes to sleep to save energy and it wakes up at the instant of the rendezvous. In the figure, after the transmission of an ACK to T_{x1}, the parent node is again ready for receiving preambles from other nodes. So, T_{x2} transmits a preamble and receives an ACK

with the same rendezvous. Preamble-clearing continues until the end of the channel polling interval of the parent node.

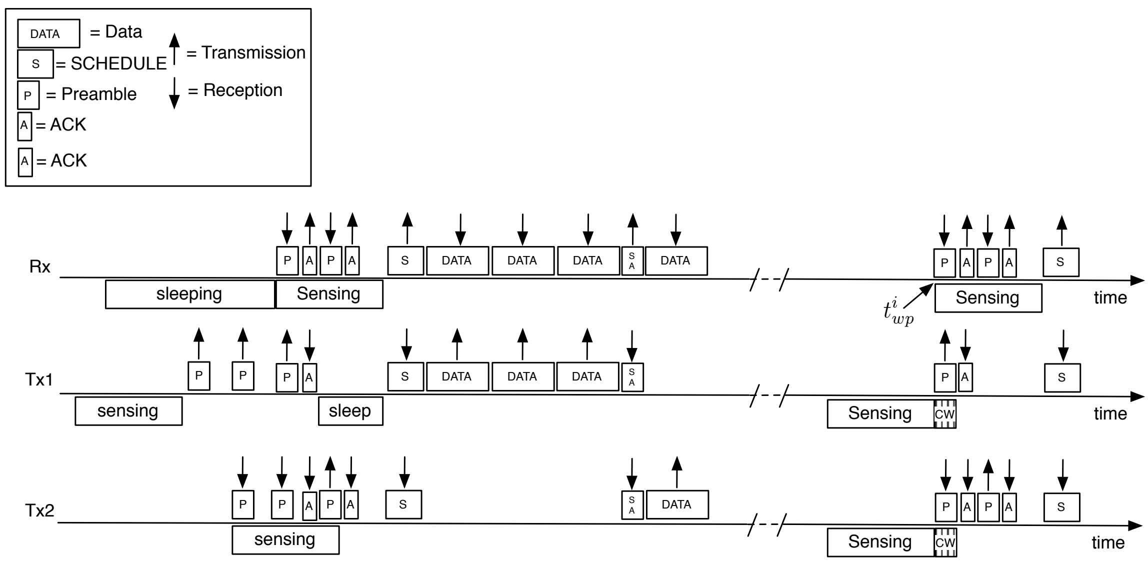


FIGURE 3.4 – LA-MAC operation during the transmission of two bursts of data frames.

***Step 4** : Broadcast of the SCHEDULE frame.*
During its channel polling period, Rx receives short preambles, clears them with ACKs, and at the end of the polling period, defines the schedule for children transmissions based on their requests priorities (cf. Sec. 3.4.3). Senders are scheduled until there is room for transmissions, that is, each receiver allocates transmission of data for an overall duration equivalent to the time remaining until its next wake-up time. At the end of polling period, Rx broadcasts the SCHEDULE frame that contains a list of instants at which transmitters can transmit their bursts :

– *Scheduled Transmitter id*, transmitters MAC addresses,
– *Schedule of node i*, the instant of the transmission end of node *i*.

Moreover, to advertise broadcast transmissions the SCHEDULE frame must also contain a flag per each scheduled transmitter ; this way all children can remain in active mode and receive broadcast radio frame.

***Step 5** : Burst transmission.*
After receiving the SCHEDULE frame, children nodes transmit their bursts at the defined instants. While waiting for their turn, a child can go to sleep and wake up at the instant of its transmission. LA-MAC protocol can support selective acknowledgments (S-ACK) [84] : after the transmission of each burst of data frames, the sender waits for an S-ACK containing the ids of correctly received data frames—if some frames are lost, they are re-transmitted during the following wake-up period of the parent node (see Fig. 3.3).

***Step 6** : Data forwarding.*

If the received data frame contains a packet to forward to the sink, the parent will take the role of child node and start sending its short preambles, immediately if the wake-up schedule of the parent is unknown, or at a specific time instant, otherwise.

Moreover, parents in the hierarchy may have polling periods delayed by some offsets that allow for forwarding packet operation like in D-MAC, however nodes do not need to be precisely synchronized.

***Step 7** : Next wake-up period.*

In order to adapt the wake-up schedule of children with respect to the schedule of parents, each child needs to know two elements : the next wake up instant of the parent node t^i_{wp} and the estimated number of contending children C^i (cf. Fig. 3.5). t^i_{wp} is contained inside an ACK frame together with the rendezvous instant while the estimation of the number of children is sent in the SCHEDULE frame, because to estimate the number of its children the parent needs to gather as much preambles as possible.

Based on this information, each child node can select a different wake-up schedule depending on the desired destination : child j will start its CCA at $t^i_{wp} - t^j_l$. To avoid collisions of preambles sent by several children at this instant, if a child detects a transmission of a preamble frame, it will randomly choose a slot within a contention window of CW according to a uniform distribution.

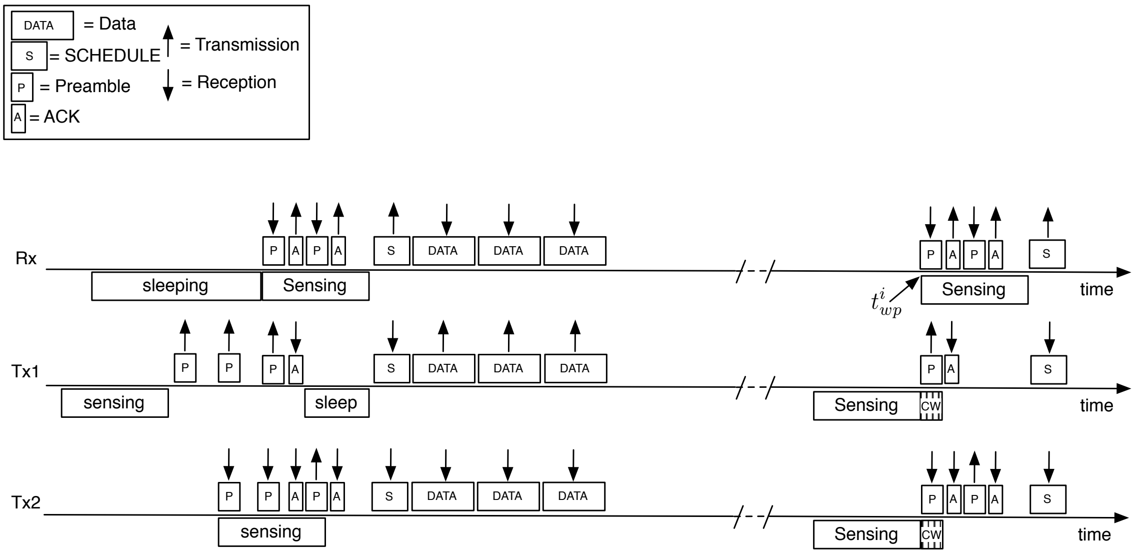


FIGURE 3.5 – LA-MAC operation during the transmission of a burst of broadcast data frames with adaptive wake-up schedule mechanism enabled.

Fig. 3.6 illustrates block diagrams of B-MAC, X-MAC and LA-MAC for comparison.

3.4.2 Broadcast Transmissions

If a node needs to broadcast a burst of data frames, it can mark the burst as broadcast in its preambles. Then, the parent node will schedule all children to be awaken during the transmission so that all nodes will receive the broadcast burst (cf. Fig. 3.5).

To allow communication from the parent to children such as for transmitting DIO routing messages [79], parent node behavior is similar to an energy saving B-MAC device, *i.e.*, it marks preambles as broadcast preambles and then it sends data after a full series of preambles ; all children that hear one broadcast preamble from a parent do not send any ACK and wait until data message to come.

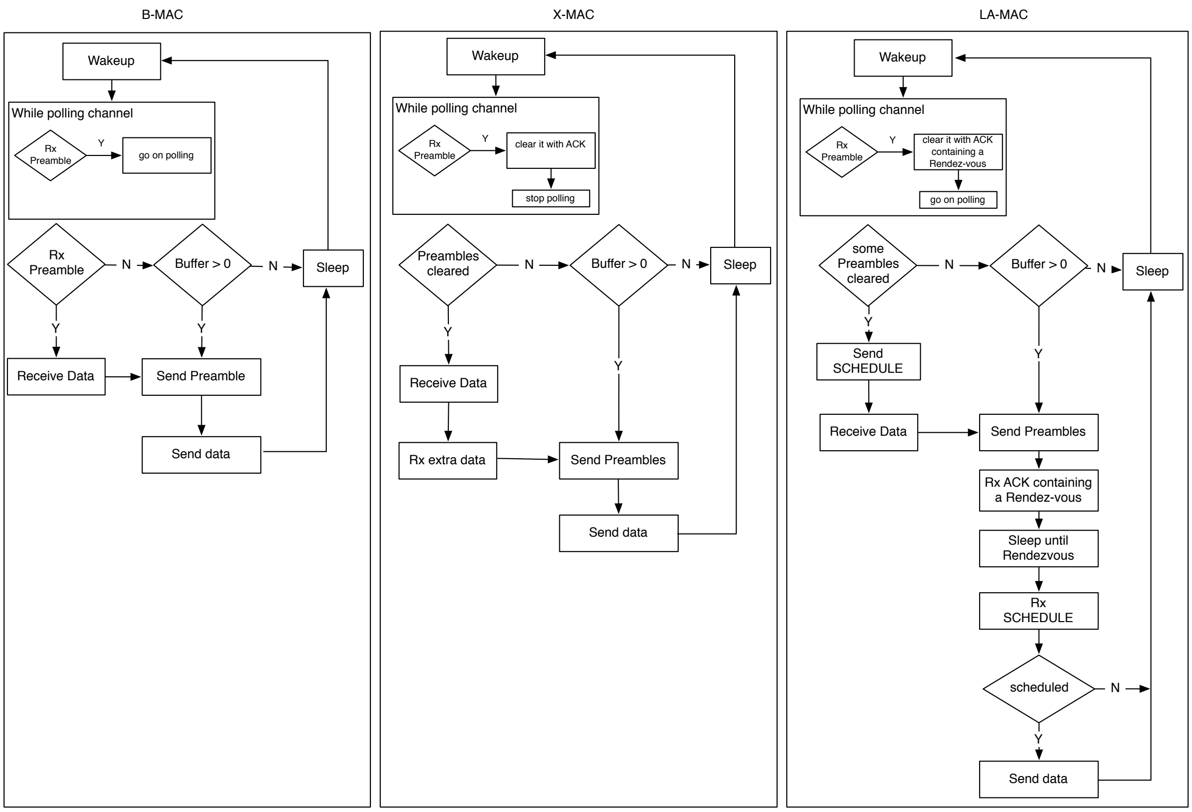


Figure 3.6 – Block diagram for B-MAC, X-MAC, and LA-MAC protocols. Diagrams refer to the case of unicast data frame transmission.

3.4.3 Traffic Differentiation

Depending on the priority specified inside a preamble frame, nodes can notify the parent node about the importance and urgency of their data frames. The parent node schedules burst requests according to the priority first, then if multiple nodes have data frames of the same priority, according to the age of frames waiting for transmission, and otherwise, all children nodes equally share channel access.

3.4.4 Frequency of Wake-Ups

Adapting wake-up interval with respect to traffic load results in energy savings [85]. With LA-MAC the frequency of wake-ups can be adapted to the traffic of nodes : if it decreases, the content of SCHEDULE packet must be increased as consequence. This means that either more users will send data frames, there are more data frames per user, or the both. In the last case, the parent node may decide to postpone its next wake-up time (decreasing its wake-up frequency). As the SCHEDULE message contains the information about the next wake-up time, all children nodes can adapt their wake-up instant too. Even if the traffic load is light, saving energy by decreasing frequency seems to be a good idea, however provided that some traffic types have also latency constraints, the frequency adaptation must be a trade-off between traffic load and QoS constraints. sends must be limited, cant be arbitrary large. Moreover, the frequency of wake-ups cannot be reduced under a given threshold. In fact, if nodes do not know the next wake-up instant of their parent (e.g. after deployment), they will need to advertise their wish to transmit by sending a train of short preambles whose length is limited. The minimal frequency of wake-ups also provides a bound of the maximum number of preambles to send to be sure to cover the wake-up interval of the intended receiver.

3.4.5 Handling Collisions

LA-MAC is collision prone when children become synchronized with respect to the same parent node and wake up at the same instant. Before the first interaction with its parent, a node is not synchronized with the wake-up period of its parent, so an average number of γ^L short preambles are needed to wake up a parent. It may happen, that two transmitters send preambles at the same time to the same receiver so that it will not be able to schedule the transmission of its children. In this case, the next time each transmitter will wake up, it will wait a random back-off time by choosing a slot within a contention window of CW according to a uniform distribution. If an ACK frame collides with another radio frame, a node may be not aware of the rendezvous for the SCHEDULE frame. In this case, the node will keep transmitting its preambles until an ACK or a SCHEDULE frame is received. If a SCHEDULE frame collides, the node cannot follow any schedule, so it will go back to sleep after a timeout and wake-up at its next wake-up instant. As data frame transmissions are scheduled by the parent, we consider that collisions of data frames are rare and we do not propose re-transmissions.

3.5 Simulation Results

We now introduce the system model used for all simulations reported in this work. Then follow numerical results (cf. Sec. 3.5.2)

3.5.1 Simulation Environment

We have implemented the analyzed MAC protocols in the OMNeT++ simulator (cf. Appendix A) for numerical evaluation. We now list major simulation details that characterize simulation scenarios :

We assume that all devices are equipped with the CC1100 [86] radio stack with bitrate of 20 Kbps. The values of power consumption for different radio modes are specific to the CC1100 transceiver considering two AA batteries that supply a voltage of 3 V. In Table 3.1 are shown current values for CC1100 sensor nodes at 0 dBm of transmission power with a central frequency of 868 MHz.

Consumed energy is evaluated only taking into account the consumption of the radio communication equipment of the nodes.

Mode \ Consumption	Current
Transmission	16.9 mA
Reception	16.4 mA
Idle	16.4 mA
Sleep	39.3 μA

Table 3.1 – CC1100 transceiver current consumption.

Simulation results are always averaged over a number of independent runs and in figures we show the corresponding confidence intervals at 95% confidence level.

With radio chipset CC1100 resulting range is about 42 m, while interference range is set as 3 dBm lower than sensibility, resulting in about 53 m.

Performance criteria involve both QoS (*i.e.*, latency and delivery ratio) and generic network criteria such as average energy consumption :

- **Latency [s]**, the average delay between the instant of packet generation and the instant of packet reception at the given sink. For broadcast messages, we also evaluate the average end-to-end delay between the generation of a broadcast message and ACK reception.
- **Access Delay [s]** Average access delay measures the time that a frame waits in MAC buffer before being sent in air.
- **Latency Per Rank of Generator [s]**, in multi-hops networks messages must be relayed from originator to the sink resulting in higher latency for messages generated by nodes far away (in number of hops) from the sink. Different MAC protocols result in different trends of latency rise with respect to the distance of originator.
- **Delivery Ratio [%]**, the ratio of the number of received packet by the sink to the total number of generated packets.
- **Delivery Ratio Per Rank of Generator [%]**, in multi-hops networks with convergecast traffic, observing the ratio of correctly delivered packet vs. the distance of packet source helps to investigate the performance of regions that are far from the sink.
- **Packet Drop Ratio [%]**, if MAC buffer size is limited and traffic congestion is high some packets coming from upper layers are discarded at MAC layer.
- **Energy Consumption [Joules]**, the energy consumed by a node due to radio activity. It can be an average value over the entire network or separated per node type (sink, sender, relay).
- **Time Spent in a Given Radio Mode [%]**, the percentage of the entire simulation time that devices spend in each radio mode.

3.5.2 Numerical Results

All simulations consist of two phases : during the first, the network is flooded with ETX probes [87] and routes toward the sinks are built according to RPL (Routing Protocol for Low power and Lossy Networks) [79] to structure the topology as a DODAG (Destination

Oriented Directed Acyclic Graph). Both ETX probes and RPL route discovery messages are broadcast. During the first phase no application layer messages are generated. In the second phase, nodes generate application layer traffic, so that application traffic and route maintenance messages must coexist in the network. The parameters for the network layer are : default RPL trickle duration equal to 8 ms and ETX probes period equal to 1 s.

We show results for one or two classes of application layer traffic ($M \in [1, 2]$) : T_{r1}, periodic *monitoring* traffic, and T_{r2}, sporadic event-driven bursty *alarm* traffic with possible high variable intensity. All alarm messages have higher priority than monitoring messages. All devices except the sinks can periodically generate alarms and monitoring messages with a rate that is dependent on the specific scenario. The sinks do not generate application traffic but only generates broadcast messages to build and maintain RPL routes.

We organize simulation results in two parts : part 1, "Large and Dense Network Simulations" reports on protocol performance with large and complex networks with high density of nodes. Part 2, "Localized Simulations" focuses on analyzing the network region that is close to the sink. The simulated version of LA-MAC does not include selective data acknowledgments to evaluate worst case conditions.

We compare the LA-MAC performance with two MAC protocols : B-MAC with a Contention Window [13, 17] and X-MAC [15]. We have chosen B-MAC with Contention Window, because it is does not require device synchronization and X-MAC, because it is energy efficient and can adjust protocol parameters to take into account changing network conditions. We do not compare LA-MAC with synchronous MAC methods, because they do not fit our requirements.

3.5.2.1 Part 1 : Large and Dense Network Simulations

We present simulation results for the case of a single sink. Simulation results are averaged over 10 runs, we compute 95 % confidence intervals to show the reliability of these 10 runs.

During a given simulation run of 10000 s, r_m is the same for all devices, but they generate packets at some random instants.

The parameters for LA-MAC are the following : channel polling duration is $t_l = 25\,ms$, the interval between two polling periods is 250 ms. The contention window of B-MAC is 32 slots. B-MAC and X-MAC have the same wake-up up period of 250 ms.

Scenario 1 : Heterogeneous Traffic.

In this scenario, communication is multi-hop and devices can act as sensors as well as relays. During each simulation run of 100000 s, r_m varies within a very large range $r_m \in [0.002, .., 0.1]$ pps. In our network setting, $N = 100$ nodes are deployed in a grid of 10x10 with the sink located in one of the corners in order to achieve a network with very large number of hops (the farthest device from the sink has rank 18). All devices generate periodical monitoring packets except one node (that is, $A = 1$) that generates both monitoring and bursts of alarm frames. Size of burst of alarm messages is $b = 20$. We present results for the case in which the alarm generator has rank equal to 10. MAC buffer size is unlimited, so messages are never dropped at the MAC layer. Fig. 3.7 shows the delivery ratio for both types of traffic while the average latency is depicted in Fig. 3.8.

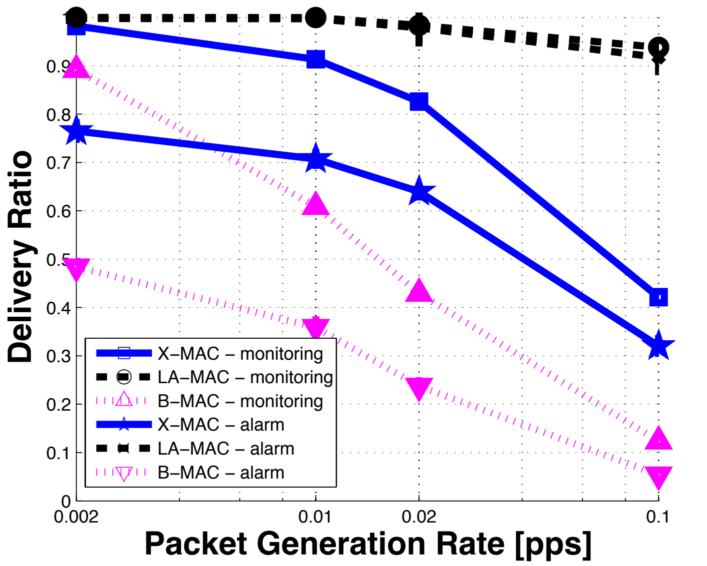


FIGURE 3.7 – Scenario 1, grid network with heterogeneous traffic. Delivery ratio of alarm and monitoring frames vs. packet generation rate. 100 nodes are deployed in a 10x10 grid ; alarm source has rank 10.

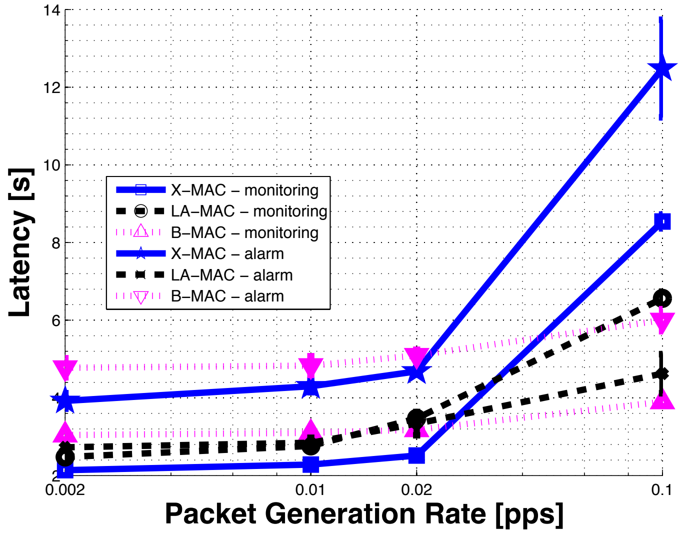


FIGURE 3.8 – Scenario 1, grid network with heterogeneous traffic. Latency of alarm and monitoring frames vs. packet generation rate. 100 nodes are deployed in a 10x10 grid, alarm source has rank 10.

In Fig. 3.9, we show the average energy consumption per node versus traffic load. LA-MAC always outperforms other protocols interdependently of traffic load. LA-MAC not only provides lower average energy consumption but it also results in the most homogeneous consumption among nodes with different ranks as presented in Table 3.3 and commented below.

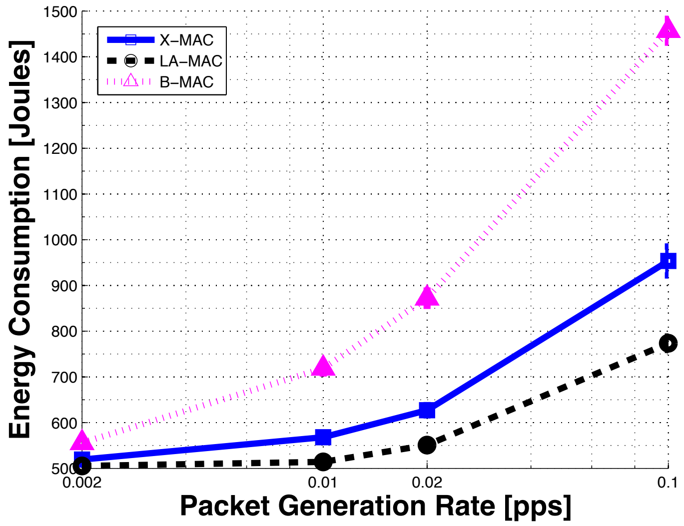


Figure 3.9 – Scenario 1, grid network with heterogeneous traffic. Energy consumption per node vs. packet generation rate. 100 nodes are deployed in a 10x10 grid, alarm source has rank 10.

In Fig. 3.10, we plot percent time that each node has spent in each radio mode (numerical values of the plot are presented in Tab. 3.2). As expected, increasing traffic load results in decreasing sleeping time, for all protocols. High transmission time of B-MAC is due to long preamble transmission. In B-MAC, idle time decreases with increasing traffic load. Reason for this is the higher number of preambles that congestion the network. When a node wakes up in fact, it spends very low time polling the channel because it immediately receives a preamble and switches the radio from idle mode to receiving mode. Resulting duty cycle, *i.e.* the sum of transmission, reception and idle percent time is shown in Fig. 3.11.

In the following figures we analyze network performance with respect to the distance of nodes from the sink. The sink being placed at one corner of the playground is the root (rank 0) of a tree may be different at each simulation run depending on RPL operations. The node that has highest rank, as rank equal to 18 and the Empirical Cumulative Distribution Function (ECDF) of node ranks is shown in Fig. 3.12. As expected all nodes have rank that is higher than 0 up to 18.

Delivery ratio of monitoring messages vs rank of nodes is shown in Fig. 3.13. LA-MAC is able to handle very high traffic loads delivering almost the 100 % of messages excepting for the case of very high traffic load where nodes with rank higher than 10 deliver around the 90 % of monitoring messages (cf. Fig. 3.13d).

Other protocols suffer the effect of forwarding messages over multiple hops. Even though traffic load is very light (r_m=0.002 *pps*), with B-MAC messages that are generated by nodes that have rank higher than 3 cannot deliver all messages.

Increasing traffic load results in severe performance degradation both for X-MAC and B-MAC until the very high traffic load scenario (r_m=0.1 *pps*)where the most penalized nodes (those with rank equal to 18) deliver the 24.3 % and 1.7 % respectively.

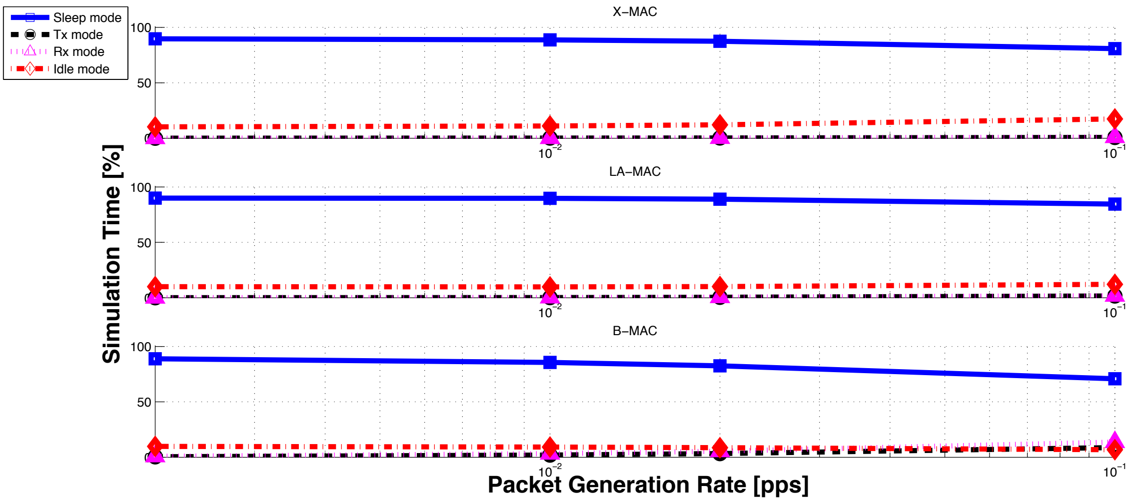


FIGURE 3.10 – Scenario 1, grid network with heterogeneous traffic. Percent ime spent in each radio mode versus the traffic load.

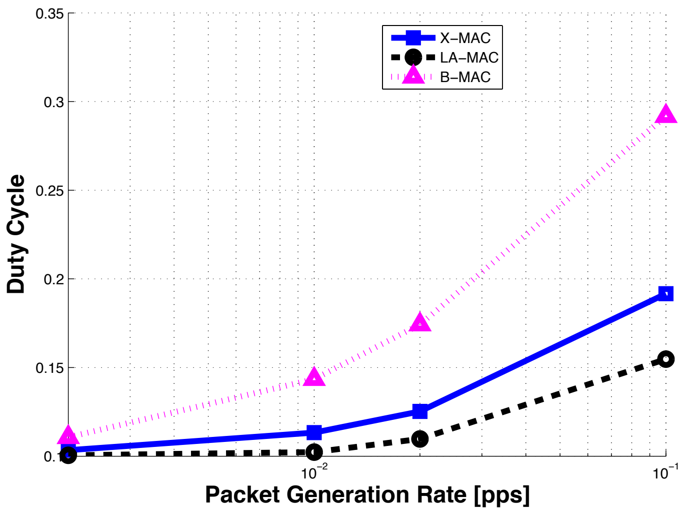


FIGURE 3.11 – Scenario 1, grid network with heterogeneous traffic. Duty cycle vs. traffic load.

Latency vs rank of nodes is shown in Figs. 3.14. With very light-to light traffic load (r_m=0.002 and r_m=0.01 *pps*) latency linearly increases with distance of message origin from the sink. When traffic load is light, LA-MAC results in an average latency that is comparable to B-MAC and both of them are higher than X-MAC. This is normal because when traffic load is light there are few messages to send, thus X-MAC results the fastest protocol.

Despite low latency, we must also consider that when traffic load is equal to 0.01 *pps*, X-MAC is not able to deliver all messages (cf. Fig. 3.13b). When traffic load is heavy, B-MAC results in lowest latency, however only very few messages are delivered to the sink

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
0.002 *pps*	0.8966	0.0002	0.0004	0.1028	0.1034
0.01 *pps*	0.8867	0.0010	0.0014	0.1108	0.1133
0.02 *pps*	0.8747	0.0020	0.0027	0.1206	0.1253
0.1 *pps*	0.8083	0.0070	0.0103	0.1743	0.1917

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
0.002 *pps*	0.8995	0.0004	0.0005	0.0996	0.1005
0.01 *pps*	0.8977	0.0018	0.0023	0.0981	0.1023
0.02 *pps*	0.8903	0.0041	0.0051	0.1006	0.1097
0.1 *pps*	0.8453	0.0158	0.0175	0.1214	0.1547

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
0.002 *pps*	0.8893	0.0047	0.0080	0.0980	0.1107
0.01 *pps*	0.8566	0.0193	0.0324	0.0918	0.1434
0.02 *pps*	0.8257	0.0332	0.0548	0.0862	0.1742
0.1 *pps*	0.7082	0.0873	0.1367	0.0678	0.2918

(c) B-MAC

Table 3.2 – Scenario 1, grid network with heterogeneous traffic. Numerical details of time spent in each radio mode versus the traffic load.

(cf. Fig. 3.13d), as a result : 12.3 % of monitoring messages are delivered with low latency. X-MAC results the protocol with highest latency when traffic is heavy, the reason for this is that when network congestion is high those monitoring messages that do not collide (42.0 %), are delivered in a best-effort way as soon as possible.

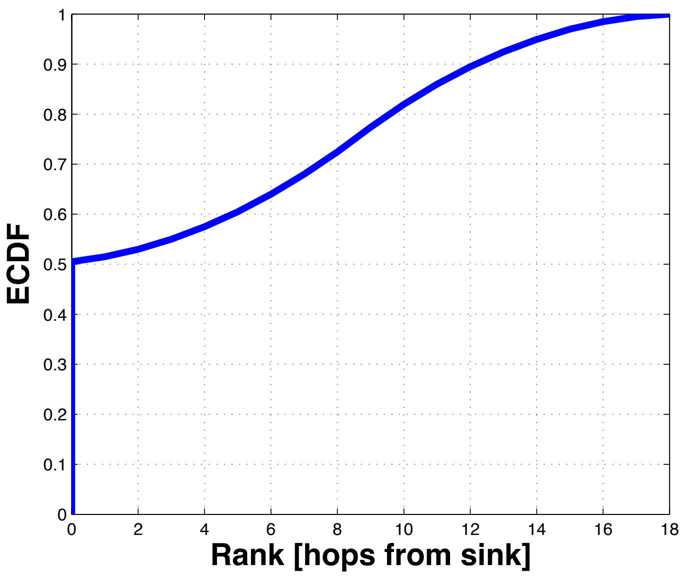
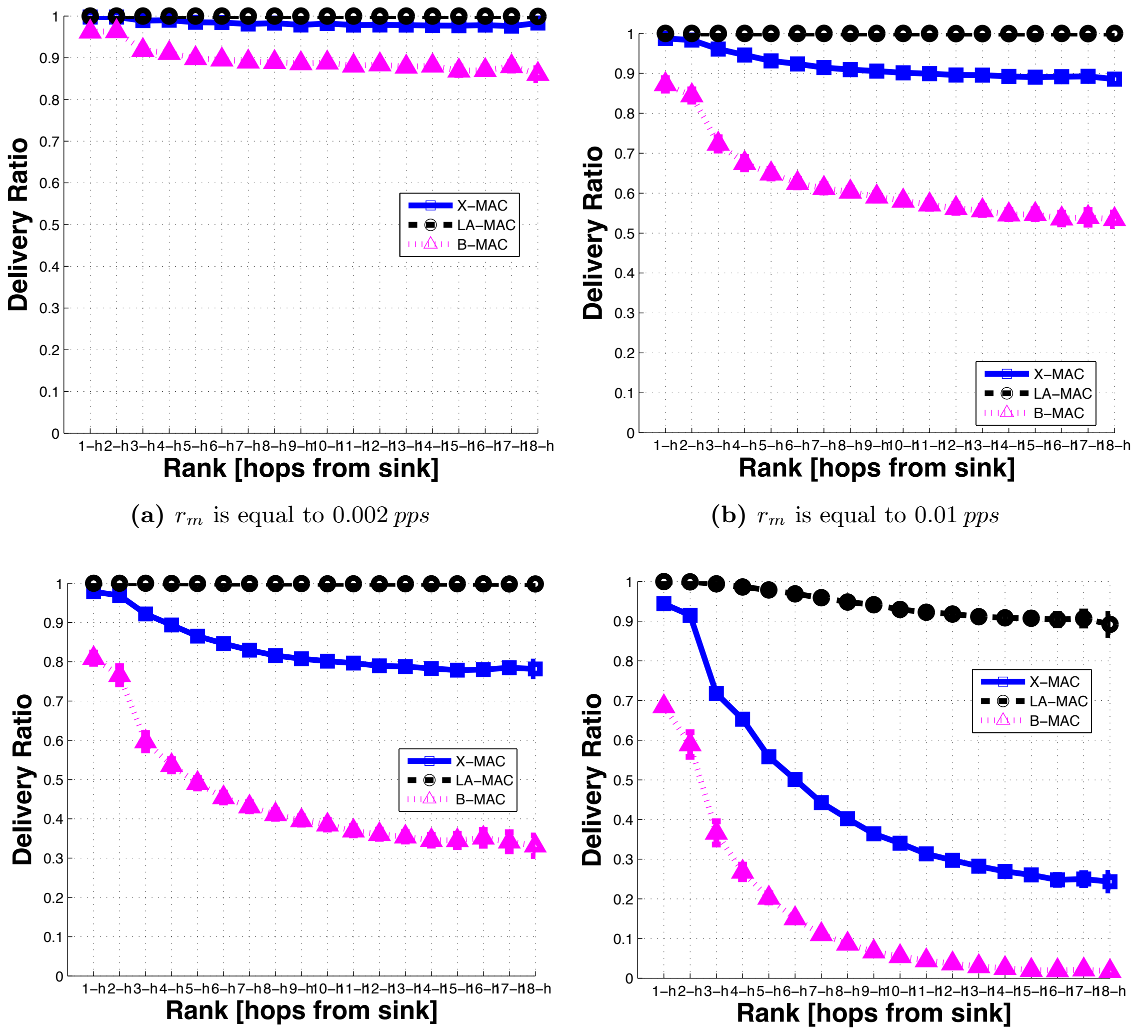


Figure 3.12 – Scenario 1, grid network with heterogeneous traffic. ECDF of the rank of nodes. There are 100 nodes are deployed in a regular 10x10 grid.



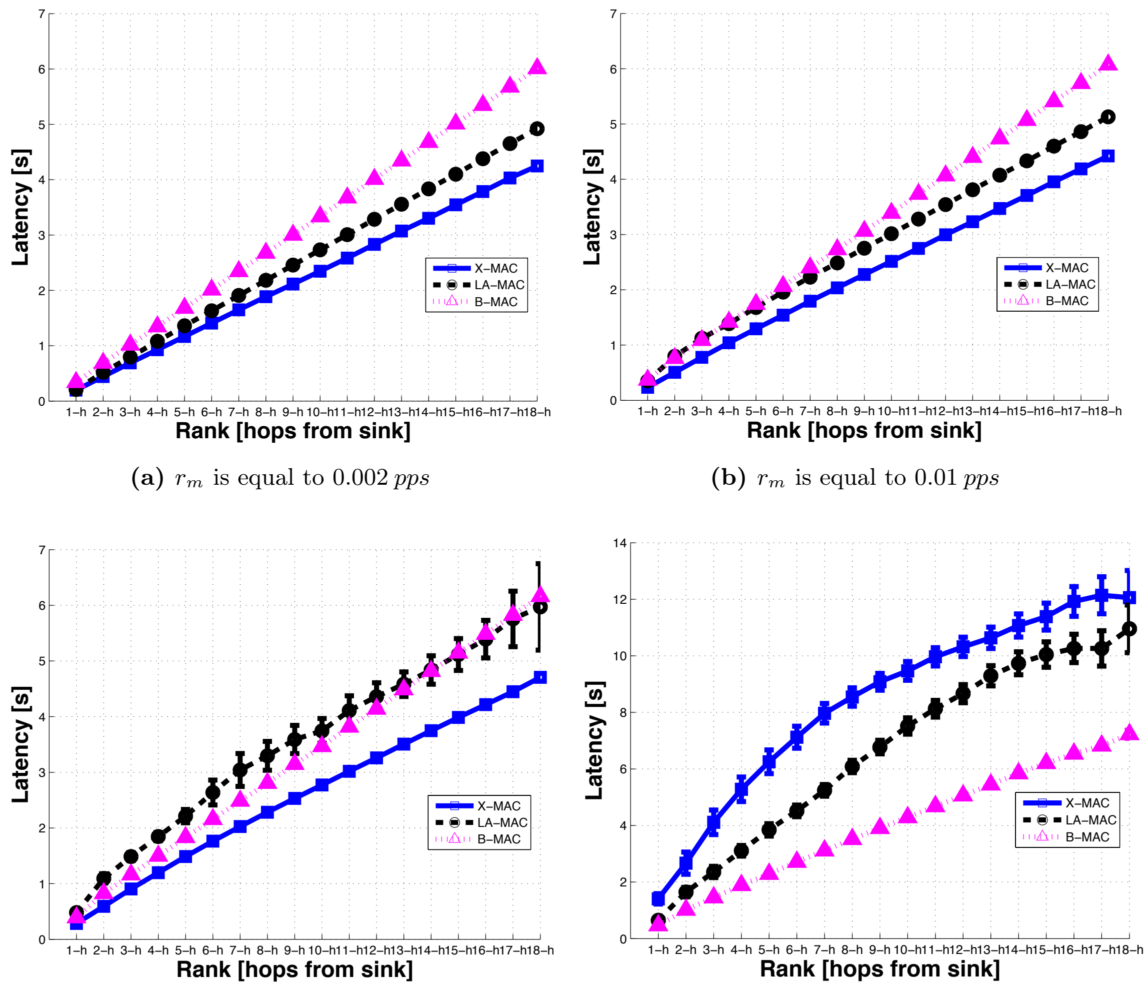
(a) r_m is equal to 0.002 *pps*

(b) r_m is equal to 0.01 *pps*

(c) r_m is equal to 0.02 *pps*

(d) r_m is equal to 0.1 *pps*

Figure 3.13 – Scenario 1, grid network with heterogeneous traffic. Delivery ratio vs. the rank of nodes. There are 100 nodes are deployed in a regular 10x10 grid, alarm source has rank 10.



(a) r_m is equal to 0.002 *pps*

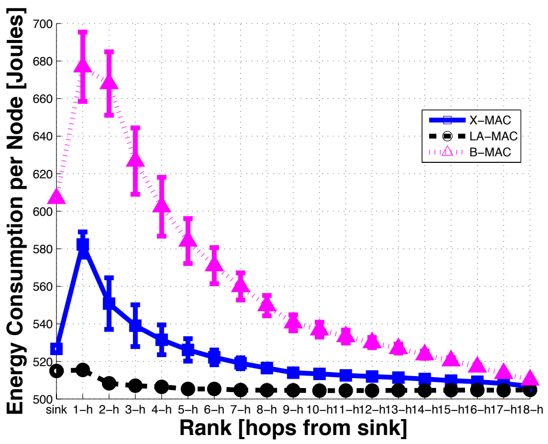
(b) r_m is equal to 0.01 *pps*

(c) r_m is equal to 0.02 *pps*

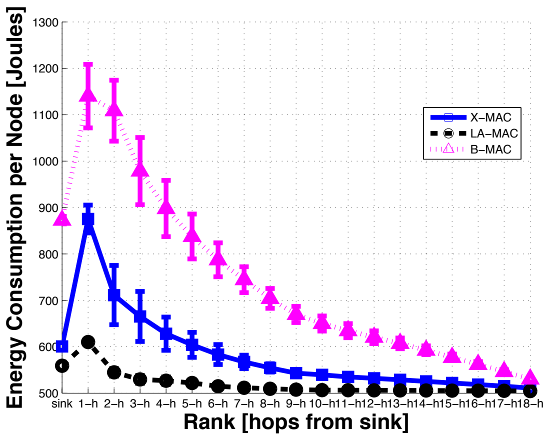
(d) r_m is equal to 0.1 *pps*

Figure 3.14 – Scenario 1, grid network with heterogeneous traffic. Latency vs. the rank of nodes. There are 100 nodes are deployed in a regular 10x10 grid, alarm source has rank 10.

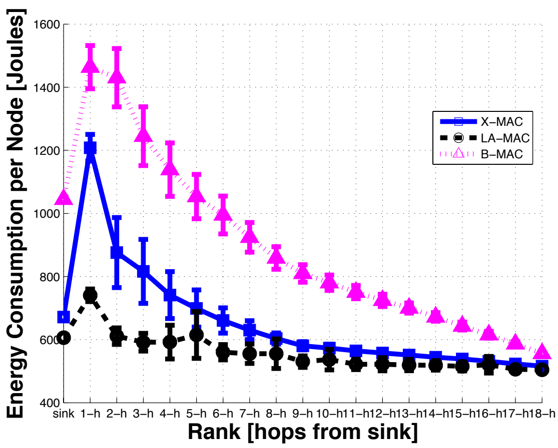
The average energy consumption per node vs rank of nodes is shown in Fig. 3.15. We observe that independently of the distance of nodes from the sink and independently of traffic load B-MAC results in the highest energy consumption. This is justified by the use of long preambles that keep radios in active mode most of the time. Independently of traffic load, nodes that are close to the sink must handle higher load than others, and this explains the decreasing trend of energy consumption with increasing distance from the sink.



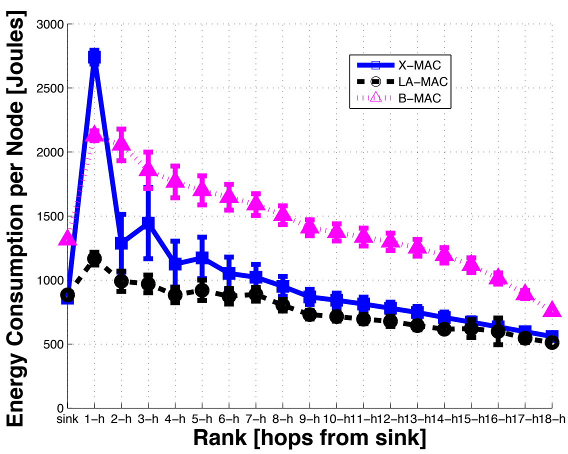
(a) r_m is equal to 0.002 *pps*



(b) r_m is equal to 0.01 *pps*



(c) r_m is equal to 0.02 *pps*



(d) r_m is equal to 0.1 *pps*

Figure 3.15 – Scenario 1, grid network with heterogeneous traffic. Energy consumption per node vs. the rank of nodes. There are 100 nodes are deployed in a regular 10x10 grid, alarm source has rank 10.

We observe that thanks to transmission of bursts, LA-MAC results in the most homogeneous energy consumption. If we define the energy spread as the difference in percentage between the node that consumes the most and the one that consumes the least, we can observe that LA-MAC always results the protocol with lowest spread as shown in Tab. 3.3.

r_m \ Protocol	X-MAC	LA-MAC	B-MAC
0.002 *pps*	0.1297	0.0205	0.2465
0.01 *pps*	0.415	0.1730	0.5348
0.02 *pps*	0.573	0.3189	0.6201
0.1 *pps*	0.7959	0.5612	0.6447

TABLE 3.3 – Scenario 1, grid network with heterogeneous traffic. Spread of average energy consumption of nodes. The spread measures the percent increment of energy consumption of the most consuming node with respect to the less consuming node.

3.5.2.2 Part 2 : Localized Simulations

In large and dense WSNs, a multitude of transmissions may occur simultaneously. The network-level performance analysis, may help us to understand what happens in simple situations such as in point-to-point communication of a couple of nodes, or in very high density situations with several children that need to send messages with different priorities to the same parent. Bad network-level performance in fact, may results from bad performance occurring in localized areas, caused by inefficient communications. We have designed several basic scenarios with the goal of investigating the performance of proposed LA-MAC protocol in simple communication scenarios. Basic scenarios that we investigate are the following :

- *Scenario 2*, point-to-point communication between one node and the sink. With such a scenario, we show the effects of varying traffic load in a communication between two nodes.
- *Scenario 3*, one hidden node interferes with a transmitting node. The presence of hidden terminals can highly influence system performance.
- *Scenario 4*, very high density network with L nodes and bidirectional traffic. Bidirectional traffic requires adaptive MAC to be efficiently handled.

Scenarios 2-3 are depicted in Fig. 3.16 while scenario 4 in Fig. 3.17.

The parameters for LA-MAC are the following : channel polling duration is $t_l = 25\,ms$, the interval between two polling periods is 250 ms. The contention window of B-MAC is 32 slots. B-MAC and X-MAC have the same wake-up up period of 250 ms. MAC buffer size is limited to 50 messages, so that in case of high congestion and inefficient MAC operation, some packets can be dropped. In scenarios 2 and 3 nodes run RPL routing protocol [79] and routes are built using ETX metric [88]. Moreover, we assume two types of application traffics (M=2) : *monitoring* and *alarms*. As a result, in all scenarios messages are bidirectional. As far as application traffic is concerned, nodes generate monitoring traffic by sending periodic packets with PGR $\in [1, 20]\,pps$. That is a very high traffic generation rate for WSNs. The alarm traffic is composed of sporadic bursts of 20 packets, probability of generating an alarm $\Gamma = 1\,\%$ and alarm timer equal to PGR. Due to the simplicity of considered scenarios and the values set for wake-up rate, we are interested in stressing

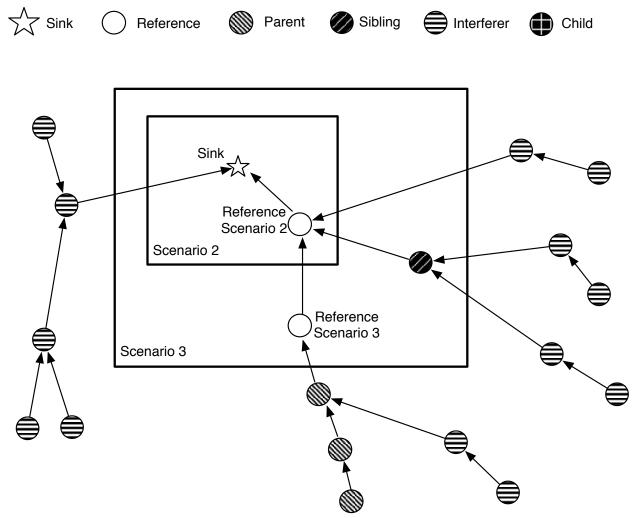


Figure 3.16 – Deployment of nodes in scenarios 2 and 3.

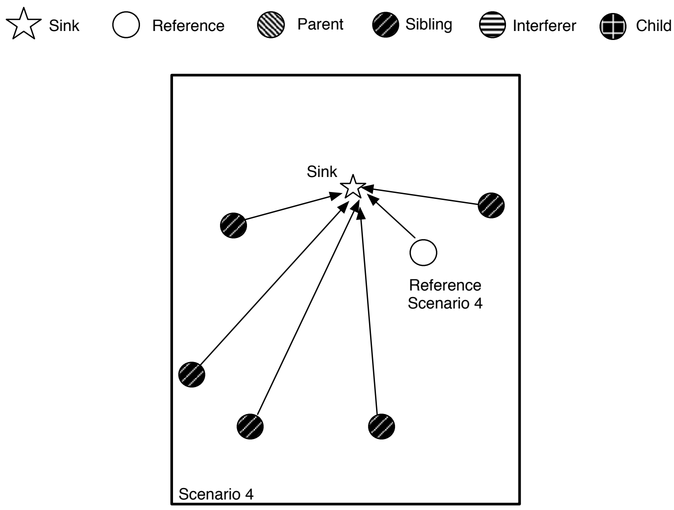
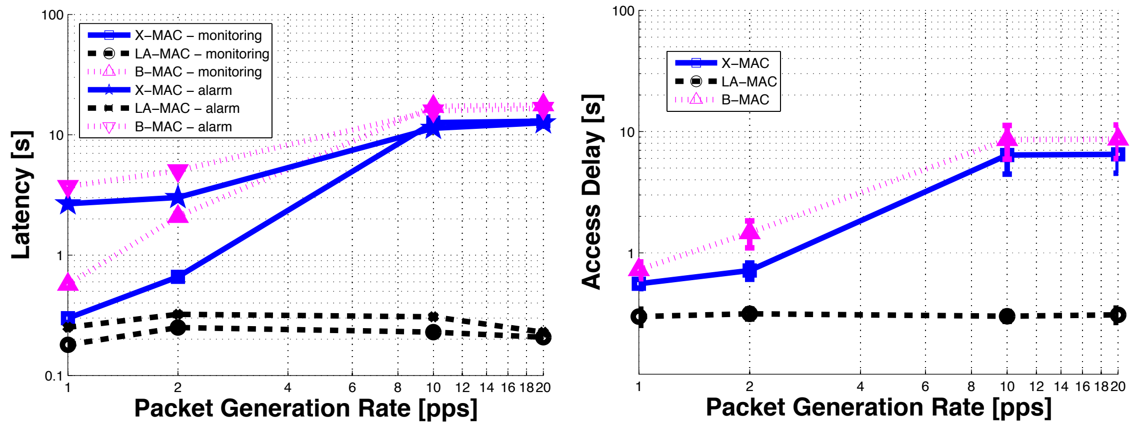


Figure 3.17 – Deployment of nodes in scenario 4.

the protocols with such traffic generation rate that spans from a value that is lower than wake-up rate until values that are much higher. Depending on the specific scenario, some nodes can generate both sources, in all cases each node can generate at most 10000 packets. We have increased the number of simulation repetitions to 20 because of the low number of nodes composing the following networks. Each run lasts 10000 s so that each node has the time to generate all the available messages (independently of the number of generated packets per second). We compute 95 % confidence intervals.

Scenario 2 : Point-to-Point Communication.
When the traffic is sporadic (the interval between two monitoring packets is larger than the

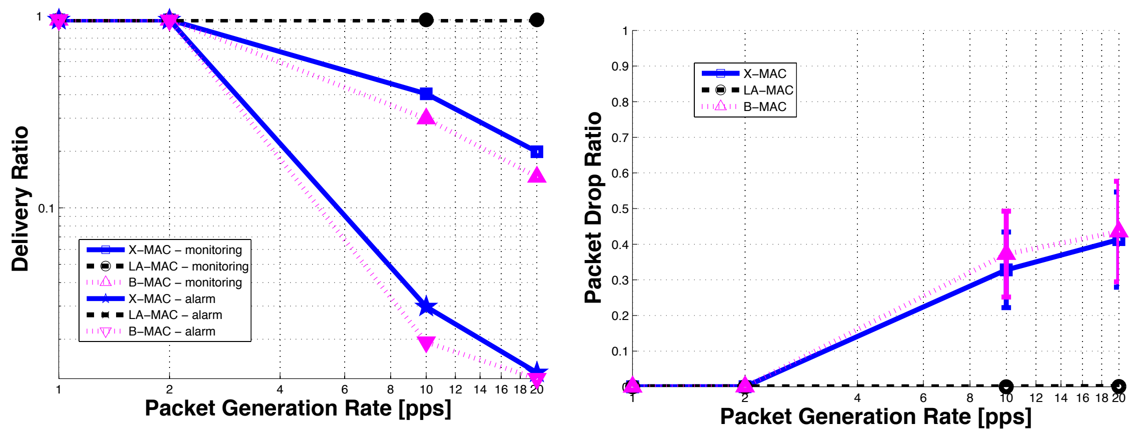
interval between two consecutive wake-up instants, *i.e.* larger than 250 ms), all protocols obtain low values of latency and access delay (see Fig. 3.18).



(a) Latency vs. traffic load. **(b)** Average access delay vs. traffic load.

FIGURE 3.18 – Scenario 2, point-to-point communication. The sender can transmit both monitoring and alarms messages.

We can observe that neither B-MAC nor X-MAC can differentiate traffic types, so when a burst of alarms enter the queue, they cannot transmit urgent packets with higher priority and the resulting latency for alarms is higher than latency for monitoring packets. As monitoring traffic increases intensity, the only protocol able to keep low latency is LA-MAC.



(a) Delivery ratio vs. traffic load. **(b)** Packet drop ratio vs. traffic load.

FIGURE 3.19 – Scenario 2, point-to-point communication. Delivery ratio and drop ratio. The sender can transmit both monitoring and alarms messages.

In this case, LA-MAC succeeds in absorbing traffic fluctuations thanks to transmission of bursts. Thanks to the simplicity of this scenario that involves one transmitter and one receiver, we can appreciate the impact of adaptive resource requests and allocations depending on local buffer size.

We can also observe that the latency levels out since the MAC packet queue is limited : when new packets arrive and the queue is filled, they are dropped (cf. Fig. 3.19b). The performance of X-MAC and B-MAC decreases with increasing traffic load resulting in very low delivery ratio (see Figures 3.19a).

Fig. 3.20 shows that average energy consumption decreases with traffic load. The reason for this behavior is that the number of messages that each node can generate is limited to 10000 and the simulation time is constant (10000 s) ; thus, if nodes generate all messages, then they only wake-up for channel polling and sleep. In the case of very light traffic load (r_m=1 *pps*), nodes spend the entire simulation time generating and transmitting messages, whereas in case of very heavy traffic load (r_m=20 *pps*), nodes finish generating their messages very soon and then they spend very long portion of the simulation time without transmitting anything, *i.e.* consuming low energy.

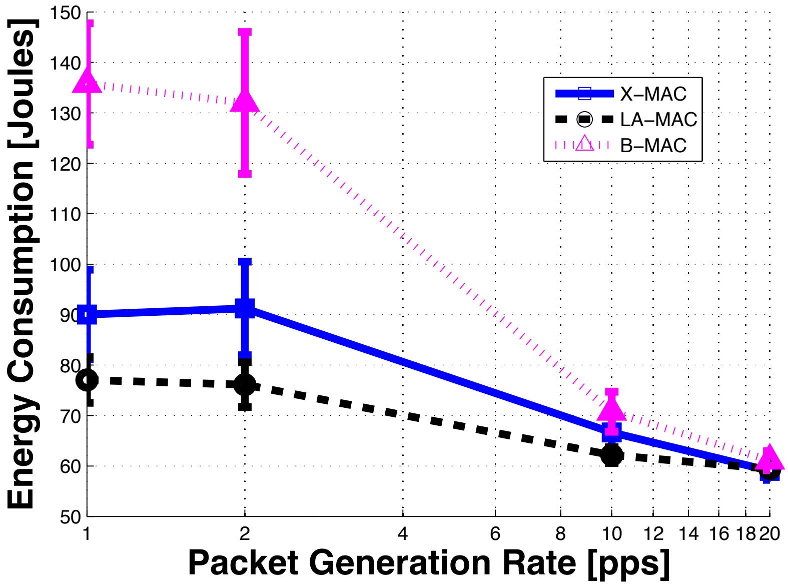


Figure 3.20 – Energy consumption per device vs. traffic load.

The details about time spent in each radio mode confirm this trend as plotted in Fig. 3.21 and also in Tab. 3.4 for a detailed analysis. In the tables, are also shown the details about duty cycle that are plotted in Fig.3.22. From the figure, we observe that the duty cycle of LA-MAC is independent of traffic load whereas it decreases for other protocols because nodes have empty buffers (most of the packets collide, see Fig. 3.19a or are dropped, see Fig. 3.19b).

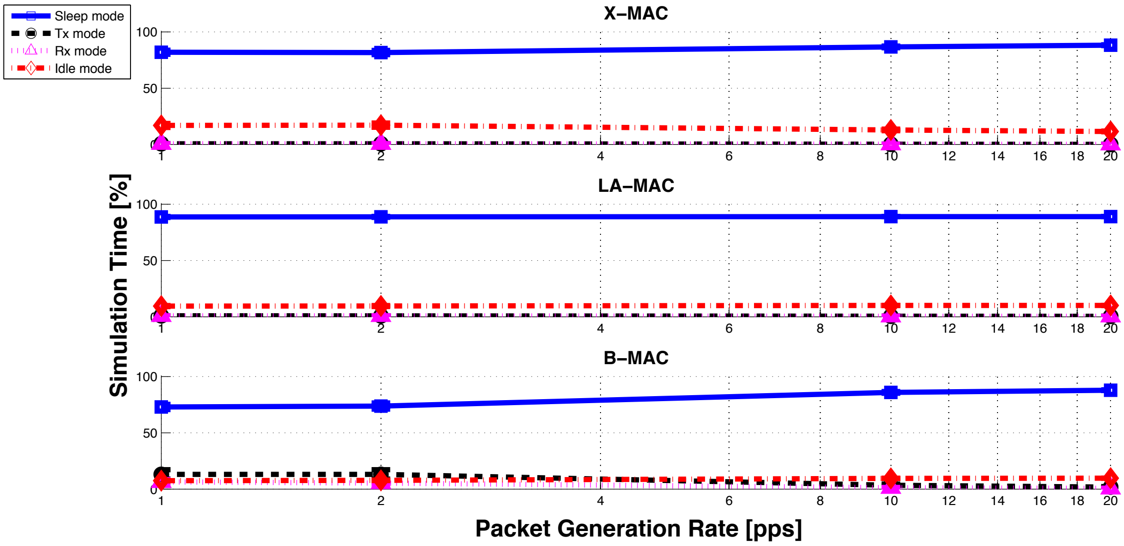


Figure 3.21 – Scenario 2, point-to-point communication. Time spent in each radio mode vs. traffic load.

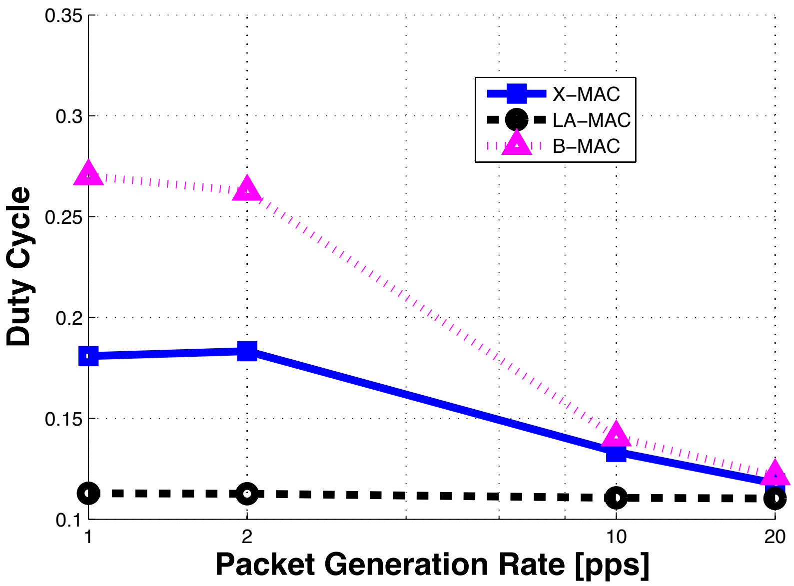


Figure 3.22 – Scenario 2, point-to-point communication. Duty cycle vs. traffic load.

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8191	0.0059	0.0060	0.1690	0.1809
2 *pps*	0.8167	0.0059	0.0060	0.1714	0.1833
10 *pps*	0.8667	0.0020	0.0022	0.1292	0.1333
20 *pps*	0.8822	0.0010	0.0012	0.1157	0.1178

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8872	0.0090	0.0085	0.0953	0.1128
2 *pps*	0.8874	0.0086	0.0081	0.0959	0.1126
10 *pps*	0.8894	0.0053	0.0047	0.1006	0.1106
20 *pps*	0.8897	0.0048	0.0043	0.1012	0.1103

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.7299	0.1309	0.0643	0.0750	0.2701
2 *pps*	0.7375	0.1308	0.0536	0.0781	0.2625
10 *pps*	0.8593	0.0341	0.0118	0.0949	0.1407
20 *pps*	0.8785	0.0176	0.0066	0.0972	0.1215

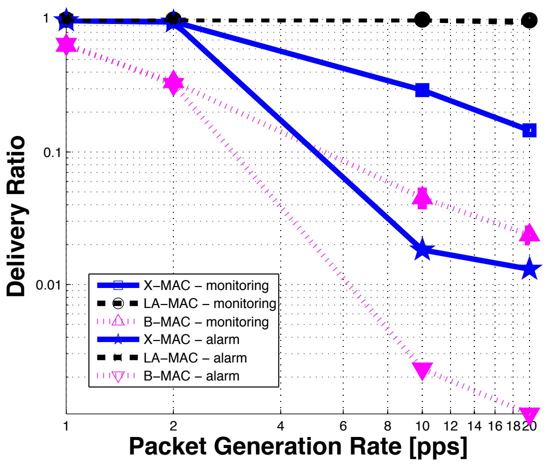
(c) B-MAC

Table 3.4 – Scenario 2, point-to-point communication. Numerical details of time spent in each radio mode versus the traffic load.

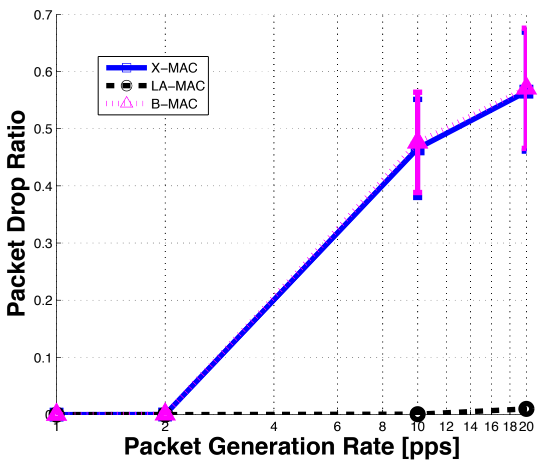
Scenario 3 : Hidden Terminal.

We consider a small network composed of three nodes : two transmitters and one receiver. As shown in Fig. 3.17, we assume that the transmitters are too far from each other so that each one cannot receive messages transmitted by the other. One transmitter sends monitoring packets while the other generates both monitoring packets and alarm bursts.

In B-MAC, if two contending nodes are hidden, they cannot overhear their preambles and a data frame collides very likely ; even though traffic load is very light, delivery ratio is lower than 100 % (cf. Fig. 3.23a). When packet generation rate is low, up to the same order as the wake-up interval (250 ms), when a node wakes up is not sure to have one message in the buffer. As traffic load increases, the probability for each transmitter to have a packet to send each time it wakes up increases as well. In the case of r_m=10 *pps*, each transmitter has an average of 2.5 waiting messages to send when it wakes up, thus, both transmitters try to wake up the sink with preambles every times they wake up. The result is a very busy channel with so many preambles that collide at the sink. When a preamble is correctly decoded by the sink, it can answer with an ACK message that allows one transmitter to immediately send data and blocks the preamble transmission of the other.



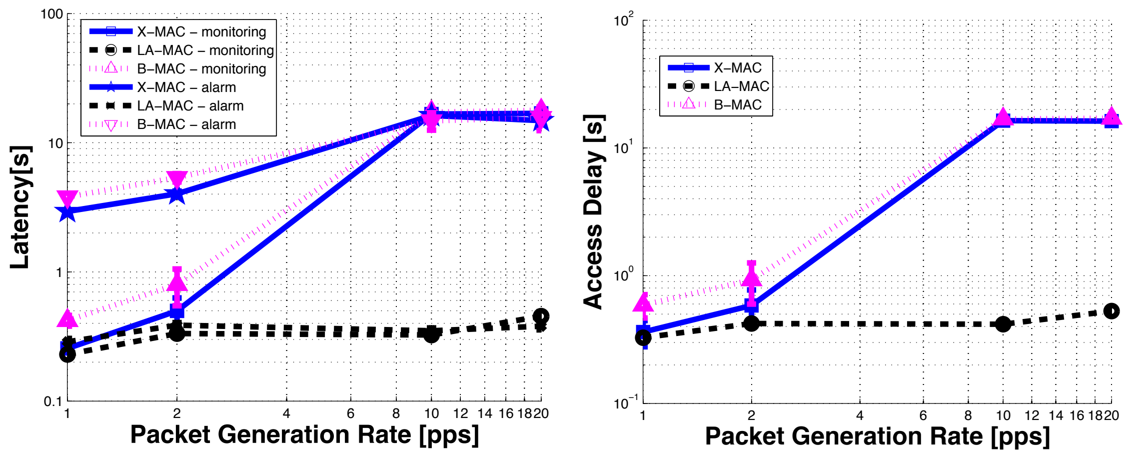
(a) Delivery ratio vs. traffic load.



(b) Packet drop ratio vs. traffic load.

Figure 3.23 – Scenario 3, hidden terminal. Delivery ratio and drop ratio. There is only one receiver and 2 senders hidden with respect to each other. Only one sender transmits alarm messages.

However, as explained in Sec. 2.2.3, in X-MAC there is a mechanism that is used to increase capacity. When a node that is sending preambles overhears an ACK sent from its parent and directed to another device, it initializes a timer with a random back-off time and directly transmits its data message when the timer fires. This mechanism is a sort of traffic load adaptation and allows the transmission of two messages per wake-up period instead of only one as in B-MAC. For this reason, the sink remains awake after the end of data reception for an extra time to receive another possible incoming packet. Even though the random back-off time is long enough to permit the first transmission to finish, if both hidden terminals have a message to send and choose a random back-off before directly transmitting their messages, there is a probability of message collisions that is not null. This is the reason for low delivery ratio of X-MAC starting from traffic load of 10 *pps* as shown in Fig. 3.23a. Moreover, when traffic load becomes high and the network is saturated, the percentage of dropped packets increases as well (cf. Figure 3.23b). In Fig. 3.24, we show that the presence of an hidden transmitter jeopardizes latency performance of both X-MAC and B-MAC. X-MAC and B-MAC provide low latency until packet generation rate is of the same order as the wake-up interval (250 ms). When packet generation rate becomes higher than wake-up interval, both protocols are not able to empty their buffers yielding high packet drop ratio, high access delay (messages wait long time in the buffer) (cf. Fig. 3.24b) and high latency. Similarly to the previous scenario, the latency of X-MAC and B-MAC levels out because of the buffer size limitation. LA-MAC presents the advantage of exploiting overhearing. At the network setup, nodes are not synchronized and their preambles may collide. However, as explained in Sec. 3.4, nodes will later on be synchronized and will randomize their access until they receive an ACK from the parent. Thus, when the first ACK or SCHEDULE frame is received, each sender will be informed of the presence of other active sibling nodes and the random back-off window will reduce the probability of collisions. As a result, LA-MAC keeps low latency and high delivery ratio (cf. Fig.3.23a) even though traffic load is heavy.



(a) Latency vs. traffic load. **(b)** Average access delay vs. traffic load.

Figure 3.24 – Scenario 3, hidden terminal. There is only one receiver and 2 senders hided one with respect to the other. Only one sender transmits alarm messages.

As in the previous scenario, average energy consumption decreases with traffic load because of the limited number of messages that nodes can generate (cf. Fig. 3.25).

The details about time spent in each radio mode are shown in Fig. 3.27. The same results along with duty cycles are presented in Tables 3.5a-3.5c. From Fig. 3.26, we observe that the duty cycle decreases with traffic load because nodes empty buffers before the end of simulation.

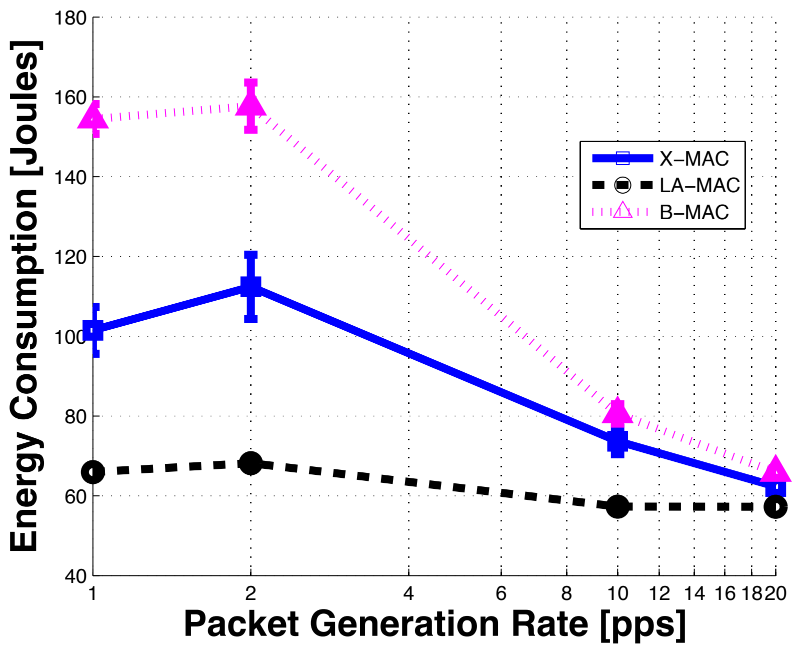


Figure 3.25 – Scenario 3, hidden terminal. Average consumed energy vs. traffic load

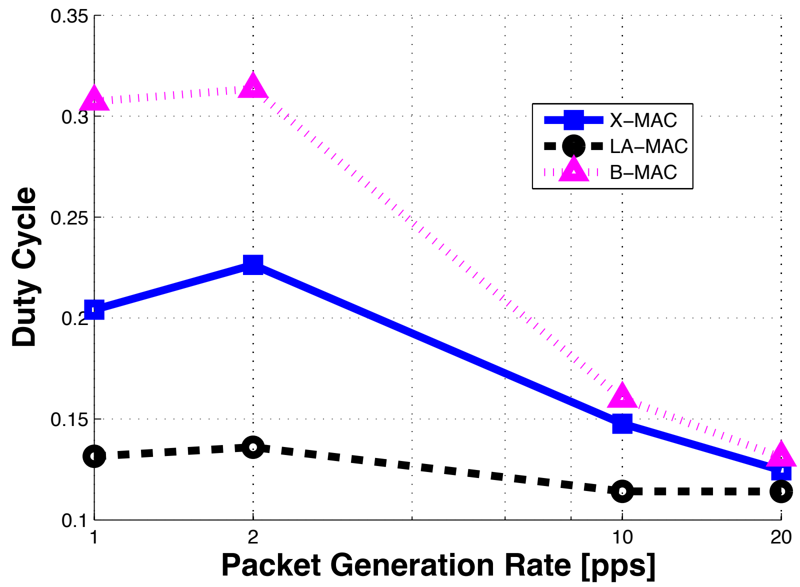


Figure 3.26 – Scenario 3, hidden terminal. Duty cycle vs. traffic load.

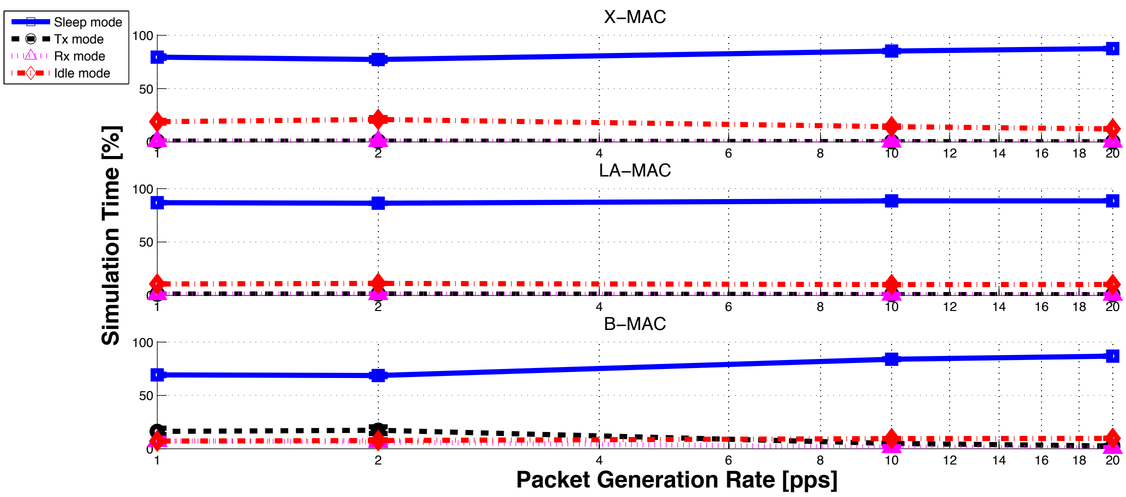


FIGURE 3.27 – Scenario 3, hidden terminal. Time spent in each radio mode vs. traffic load.

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.7959	0.0071	0.0074	0.1896	0.2041
2 *pps*	0.7737	0.0076	0.0079	0.2108	0.2263
10 *pps*	0.8523	0.0023	0.0024	0.1430	0.1477
20 *pps*	0.8754	0.0011	0.0013	0.1221	0.1246

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8685	0.0128	0.0114	0.1073	0.1315
2 *pps*	0.8639	0.0131	0.0113	0.1117	0.1361
10 *pps*	0.8859	0.0071	0.0065	0.1006	0.1141
20 *pps*	0.8859	0.0063	0.0057	0.1021	0.1141

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.6927	0.1623	0.0728	0.0722	0.3072
2 *pps*	0.6865	0.1738	0.0627	0.0770	0.3135
10 *pps*	0.8399	0.0502	0.0145	0.0954	0.1601
20 *pps*	0.8690	0.0256	0.0080	0.0974	0.1310

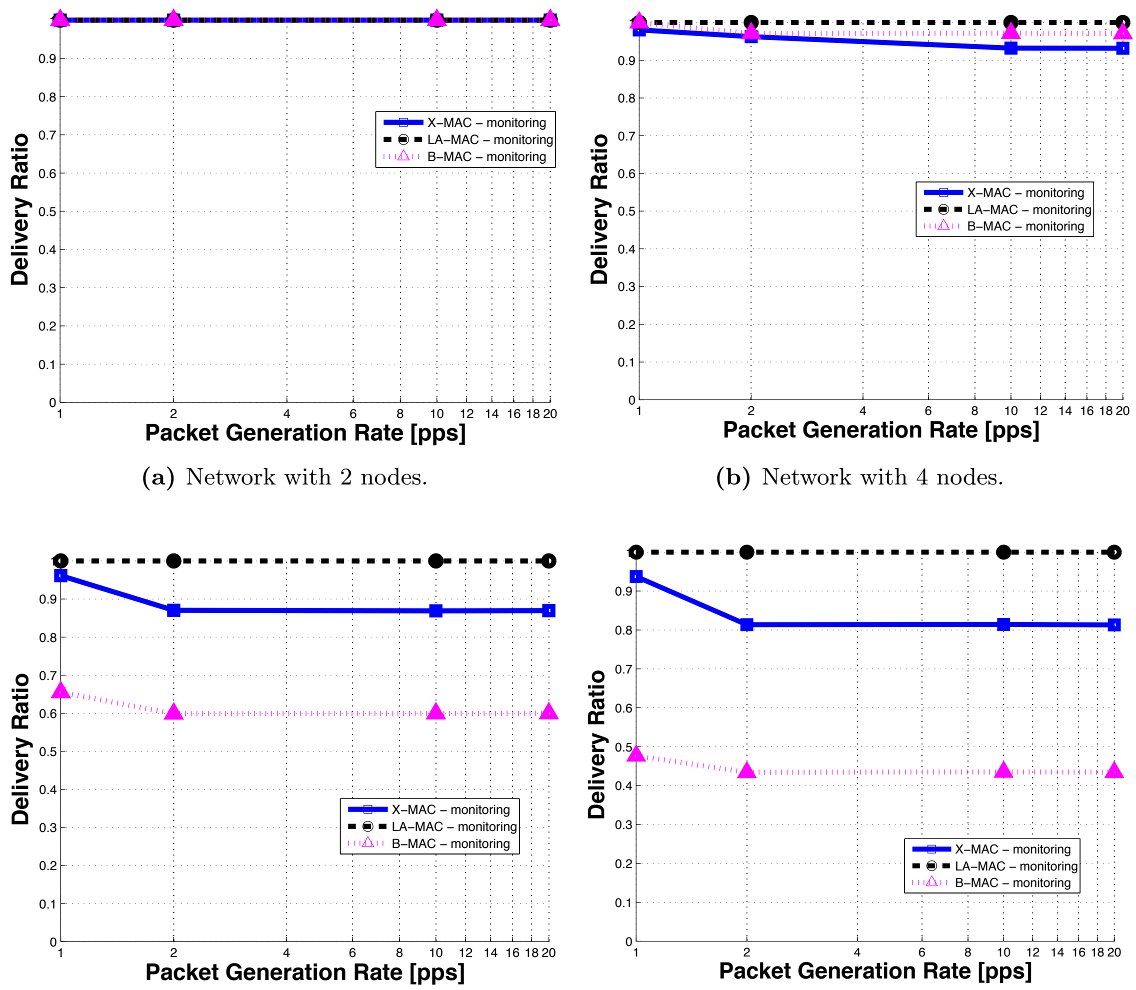
(c) B-MAC

Table 3.5 – Scenario 3, hidden terminal. Numerical details of time spent in each radio mode versus the traffic load. Last column presents the duty cycle.

Scenario 4 : Dense Star Topology.

We have set up an initial topology scenario with a variable number of senders and one sink. The number of senders varies in the range $L \in [0, 7]$ yielding a network size in the range $[1, 8]$. All devices are located within the radio coverage of each other so that increasing the number of senders also increases the number of neighbors against whom each node must contend. In this scenario, we focus on performance of high traffic load situations, therefore we consider a single application layer (M=1), that is the monitoring traffic with $r_m \in [1, 20]$ *pps*.

The resulting cumulative traffic load becomes extremely high for high PGR when the number of neighbors is higher than 2. Although such high traffic load is non realistic for WSNs, we are interested in stressing the protocols to compare their relative performance. Moreover, for the same reason, in this scenario, the buffer size is unlimited so that all



(a) Network with 2 nodes.

(b) Network with 4 nodes.

(c) Network with 6 nodes.

(d) Network with 8 nodes.

Figure 3.28 – Scenario 4, dense star topology. Delivery ratio vs. the traffic load per different network sizes.

messages entering the MAC queue are backlogged (drop ratio is always zero).

As in the previous scenarios, only one sender can transmit alarm burst while all nodes transmit monitoring messages.

In Fig. 3.28, we show the delivery ratio versus traffic load per different network sizes and in Fig. 3.29, we present the same results in a single plot. We observe that LA-MAC

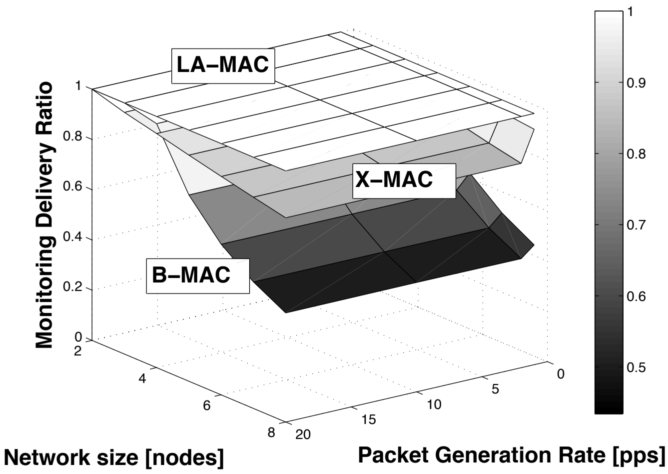
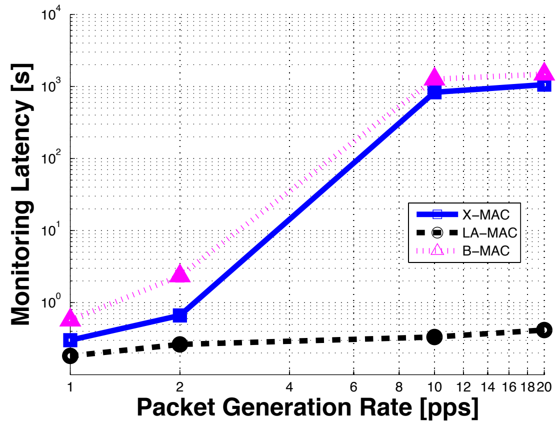


Figure 3.29 – Scenario 4, dense star topology. Monitoring messages delivery ratio vs. traffic load and network size.

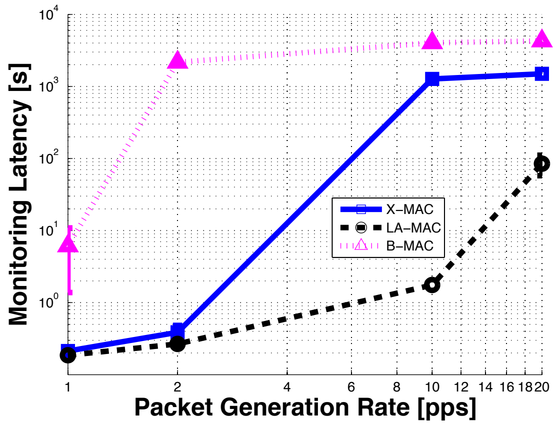
always results in 100 % of delivery ratio ; no collisions occur independently of traffic load and network size. This is the consequence of message overhearing and localized scheduling of burst transmissions. Other protocols suffer from the presence of an increasing number of neighbors. With B-MAC, there are no collisions as in LA-MAC. In fact, thanks to the long preamble each time a node wins the channel, it blocks the transmission of others that overhear a portion of the preamble and the data. In the case of all nodes in the radio range, the long preambles prevent collisions to occur.

However, the drawback of long preamble is the dramatic increase of latency and channel access even for "low" values of traffic load (Figs. 3.30-3.31b). As a consequence, at the end of the simulation, all messages that are not delivered to the sink remain in the MAC buffer. X-MAC is able to deliver more messages than B-MAC when traffic load is light, however as PGR increases with the number of neighbors several collisions occur when nodes transmit as second senders (using the extra back-off timer).

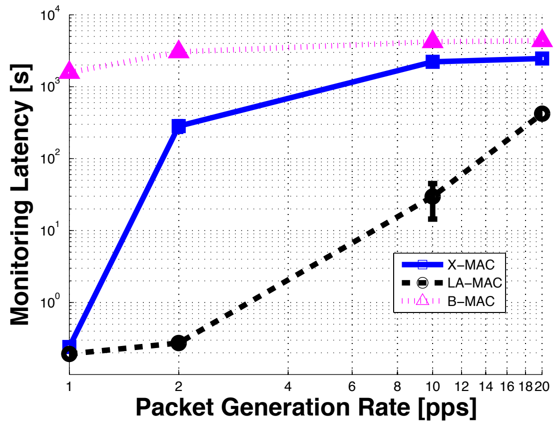
Independently of the number of neighbors, when traffic load is light (up to 2 *pp*s per node), LA-MAC provides low latency with 100 % of delivery (cf. Fig. 3.30). Packets wait short time in the buffer before being included in a burst of messages (Fig. 3.32). Then, when traffic load becomes very heavy, the average time that each node spends in the buffer exponentially increases resulting in very long latency. With B-MAC, latency and access delay are large even with few nodes and low traffic load. In fact, only one packet can be sent by each node per active period. X-MAC shows intermediate results of latency and access delay.



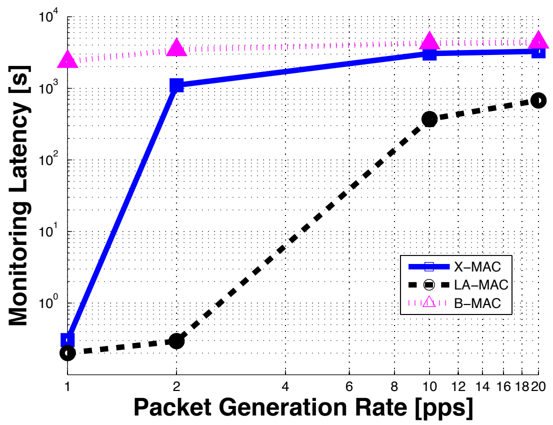
(a) Network with 2 nodes.



(b) Network with 4 nodes.

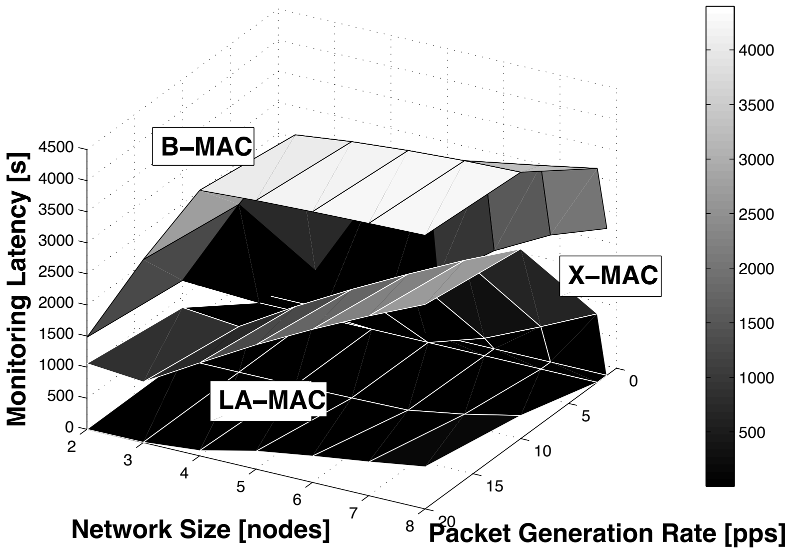


(c) Network with 6 nodes.

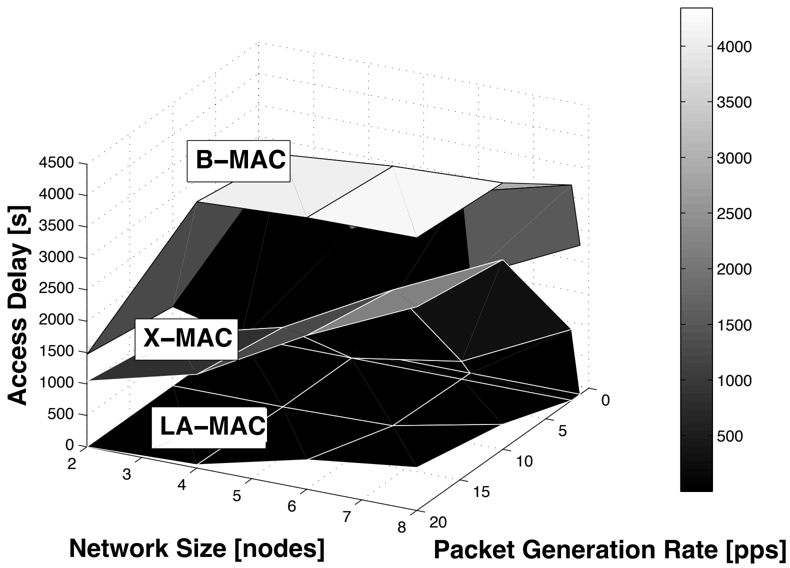


(d) Network with 8 nodes.

Figure 3.30 – Scenario 4, dense star topology. Latency vs. the traffic load per different network sizes.

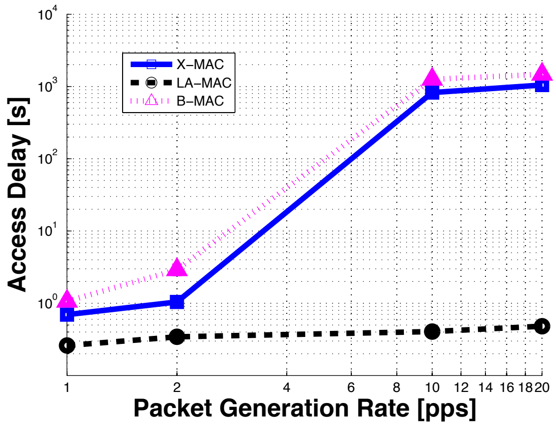


(a) Latency vs. traffic load.

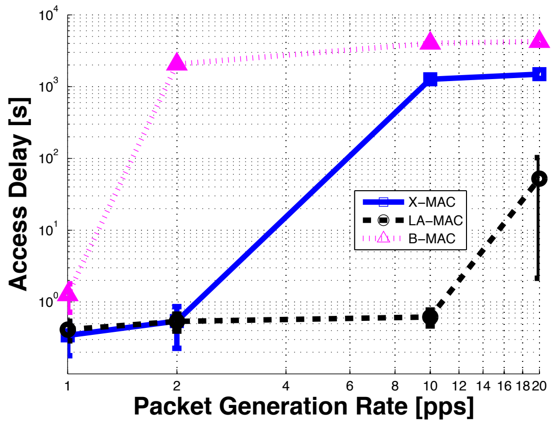


(b) Access delay vs. traffic load.

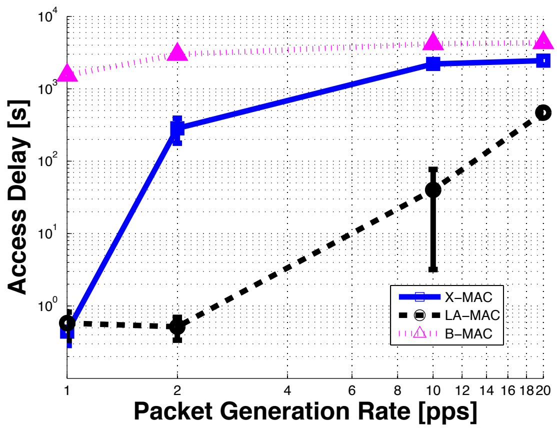
Figure 3.31 – Scenario 4, dense star topology. Latency and access delay versus the traffic load per different network sizes.



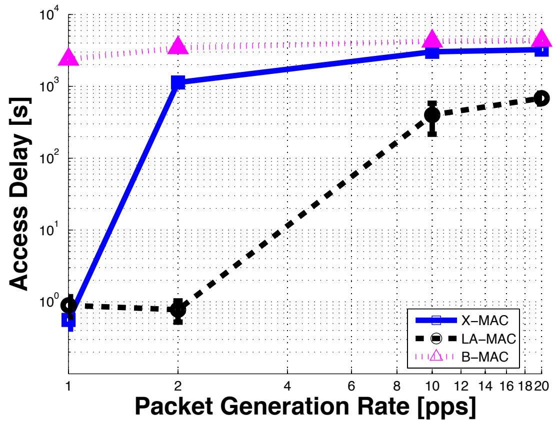
(a) Network with 2 nodes.



(b) Network with 4 nodes.



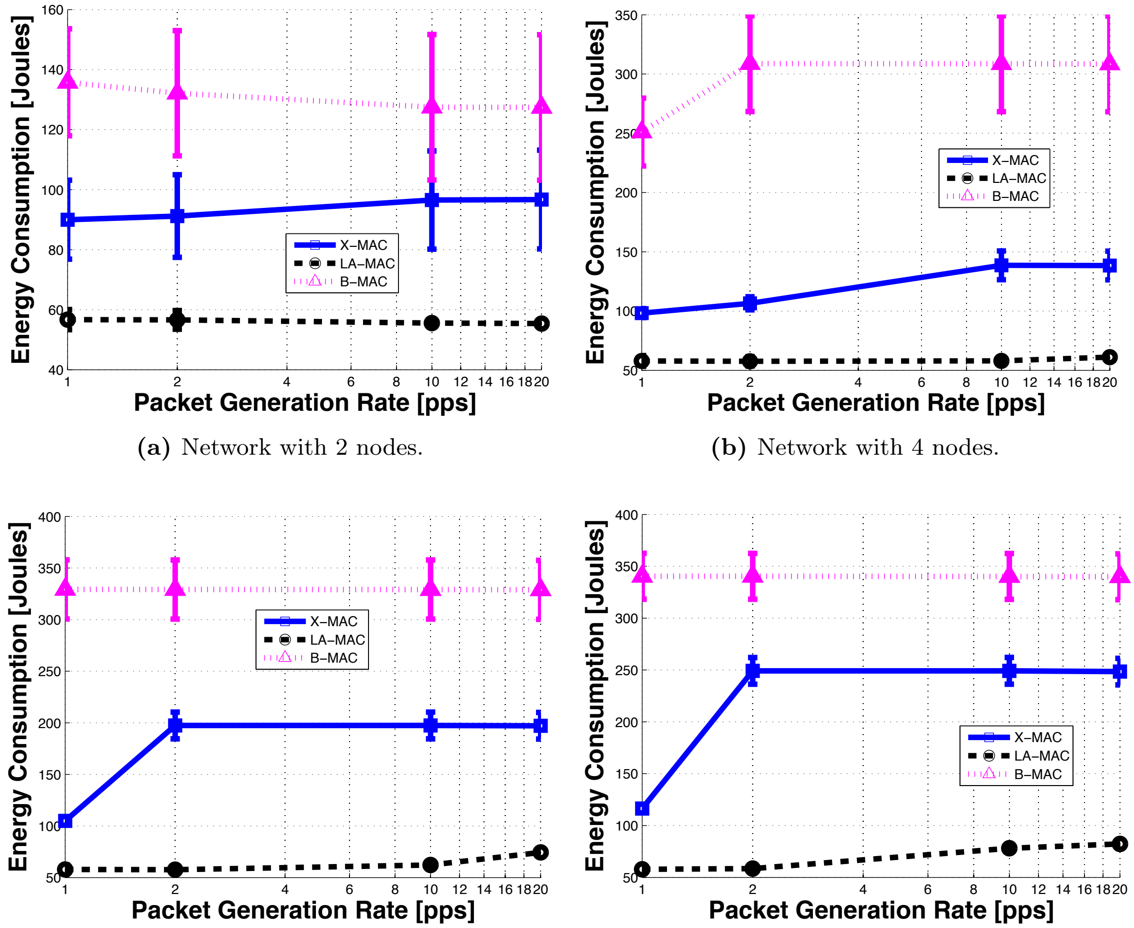
(c) Network with 6 nodes.



(d) Network with 8 nodes.

Figure 3.32 – Scenario 4, dense star topology. Access delay vs. the traffic load per different network sizes.

As far as energy consumption is concerned, we observe that all protocols level out. As expected, LA-MAC is the most energy efficient protocol, followed by X-MAC and B-MAC as shown in Figs. 3.33-3.34. As expected, increasing traffic load results in increasing duty cycle of nodes. As traffic load becomes heavy, all nodes spend much less time in sleeping mode (see Figs. 3.35a-3.36b and Tables 3.6-3.9 for numerical details).



(a) Network with 2 nodes. (b) Network with 4 nodes.

(c) Network with 6 nodes. (d) Network with 8 nodes.

Figure 3.33 – Scenario 4, dense star topology. Energy consumption versus the traffic load per different network sizes.

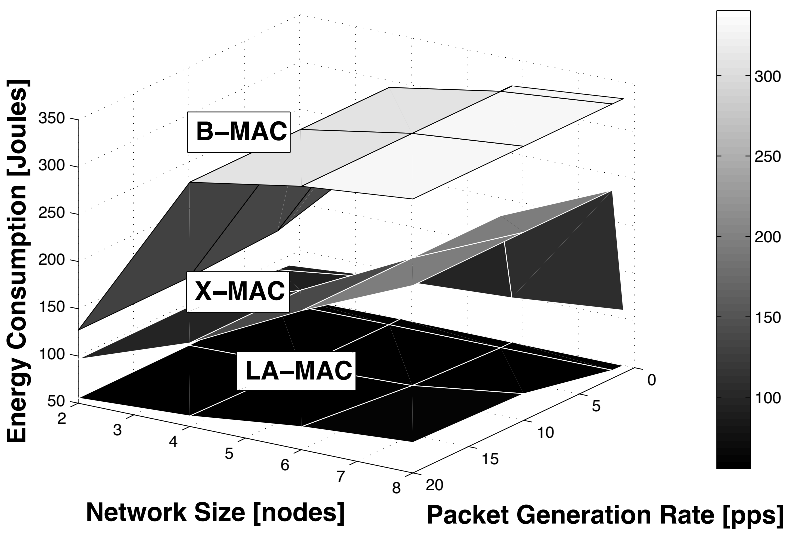
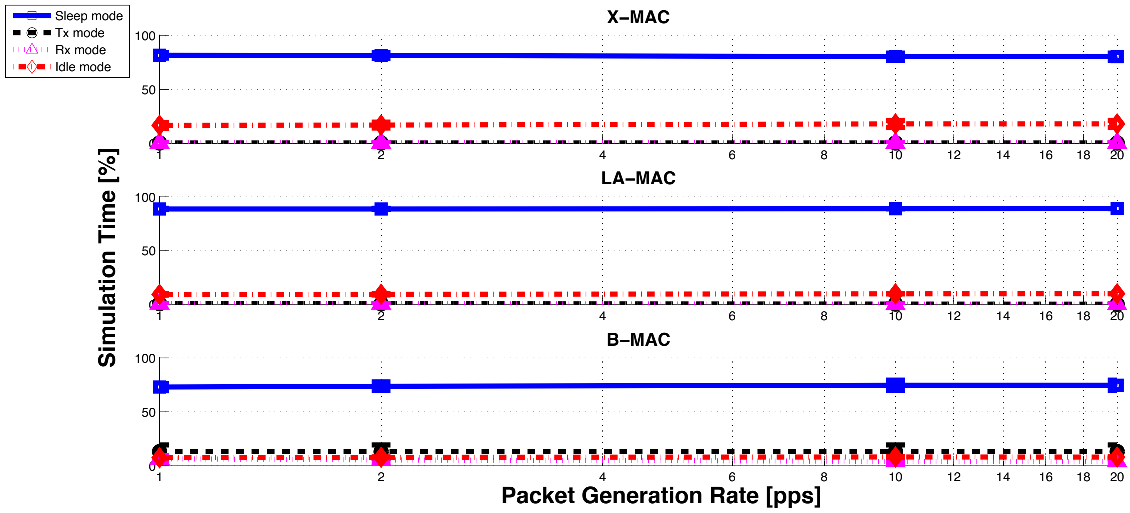
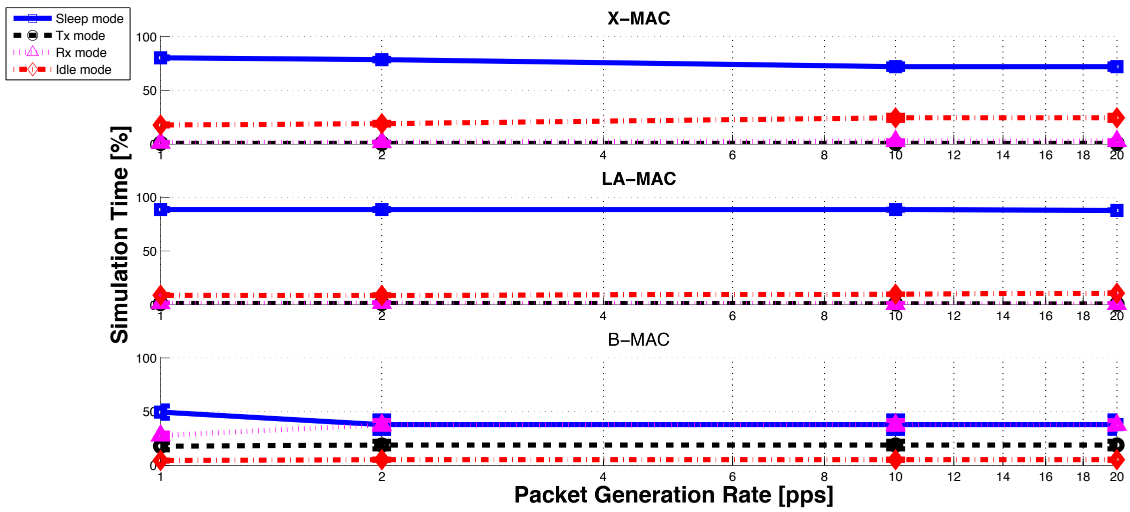


FIGURE 3.34 – Scenario 4, dense star topology. Energy consumption vs. traffic load.

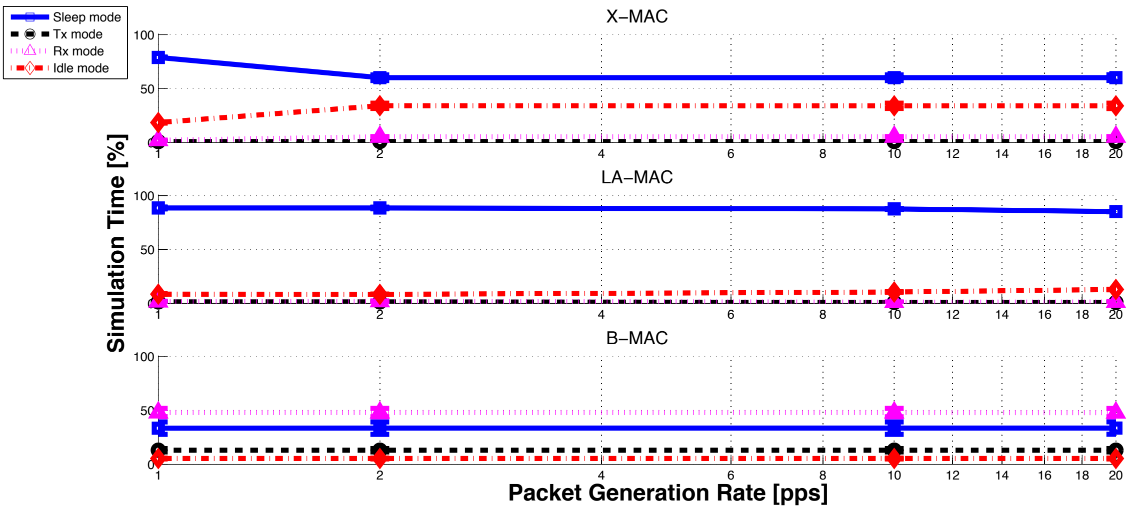


(a) Network with 2 nodes.

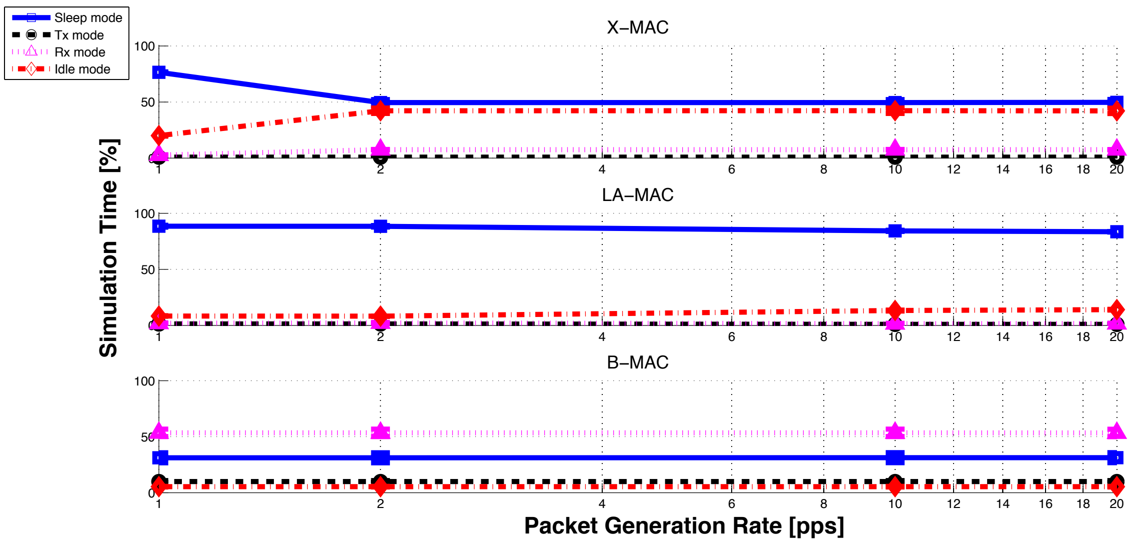


(b) Network with 4 nodes.

FIGURE 3.35 – Scenario 4, dense star topology. Time spent in each radio mode versus the traffic load per two different network sizes of 2 and 4 nodes, respectively.



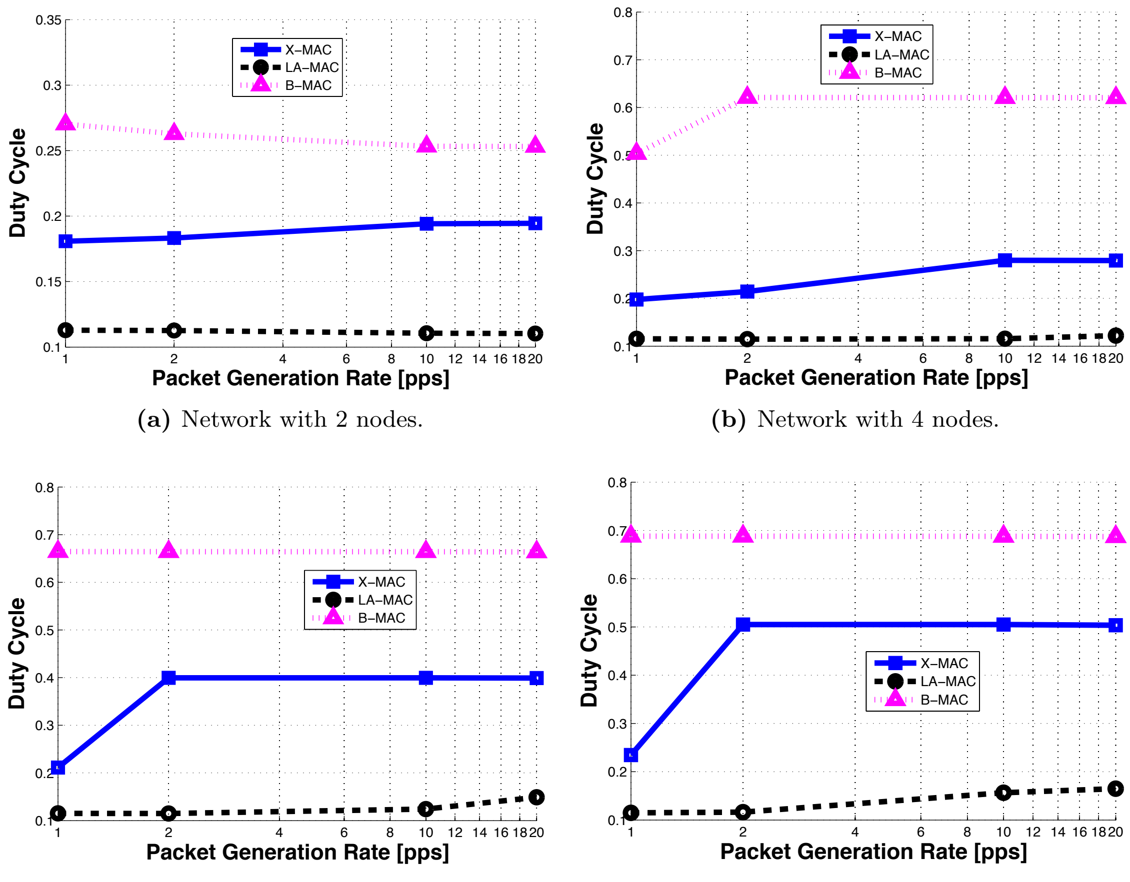
(a) Network with 6 nodes.



(b) Network with 8 nodes.

Figure 3.36 – Scenario 4, dense star topology. Time spent in each radio Mode versus the traffic load per two different network sizes of 6 and 8 nodes, respectively.

We observe that LA-MAC results in low duty cycle *e.g.* 16.5 % in the worst case of 7 senders and $r_m = 20\,pps$, whereas X-MAC and B-MAC result in respectively 50.4 % and 68.7 % in the same case (cf. Fig. 3.37).



(a) Network with 2 nodes.

(b) Network with 4 nodes.

(c) Network with 6 nodes.

(d) Network with 8 nodes.

FIGURE 3.37 – Scenario 4, dense star topology. Duty cycle vs. traffic load per different network size.

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8192	0.0059	0.0060	0.1689	0.1808
2 *pps*	0.8168	0.0059	0.0060	0.1714	0.1832
10 *pps*	0.8059	0.0059	0.0060	0.1823	0.1941
20 *pps*	0.8055	0.0059	0.0060	0.1826	0.1944

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8871	0.0091	0.0085	0.0953	0.1128
2 *pps*	0.8873	0.0087	0.0081	0.0958	0.1127
10 *pps*	0.8894	0.0053	0.0047	0.1006	0.1106
20 *pps*	0.8897	0.0048	0.0043	0.1013	0.1103

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.7298	0.1309	0.0644	0.0750	0.2702
2 *pps*	0.7372	0.1309	0.0539	0.0780	0.2628
10 *pps*	0.7467	0.1309	0.0414	0.0810	0.2533
20 *pps*	0.7468	0.1309	0.0413	0.0810	0.2532

(c) B-MAC

Table 3.6 – Scenario 4, dense star topology. Numerical details of time spent in each radio mode versus the traffic load. Network size is 2 nodes.

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8024	0.0074	0.0127	0.1775	0.1976
2 *pps*	0.7856	0.0076	0.0168	0.1901	0.2144
10 *pps*	0.7202	0.0072	0.0278	0.2448	0.2798
20 *pps*	0.7206	0.0072	0.0277	0.2445	0.2794

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8847	0.0124	0.0137	0.0892	0.1153
2 *pps*	0.8853	0.0122	0.0149	0.0875	0.1147
10 *pps*	0.8844	0.0087	0.0078	0.0991	0.1156
20 *pps*	0.8781	0.0073	0.0073	0.1074	0.1219

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.4965	0.1793	0.2783	0.0460	0.5035
2 *pps*	0.3792	0.1898	0.3773	0.0536	0.6208
10 *pps*	0.3793	0.1898	0.3772	0.0536	0.6207
20 *pps*	0.3797	0.1898	0.3769	0.0537	0.6203

(c) B-MAC

Table 3.7 – Scenario 4, dense star topology. Numerical details of time spent in each radio mode versus the traffic load. Network size is 4 nodes.

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.7888	0.0078	0.0189	0.1845	0.2112
2 *pps*	0.6003	0.0079	0.0520	0.3397	0.3997
10 *pps*	0.6005	0.0079	0.0520	0.3396	0.3995
20 *pps*	0.6010	0.0079	0.0519	0.3392	0.3990

(a) X-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8849	0.0130	0.0163	0.0858	0.1151
2 *pps*	0.8851	0.0129	0.0184	0.0836	0.1149
10 *pps*	0.8759	0.0083	0.0101	0.1056	0.1241
20 *pps*	0.8511	0.0087	0.0119	0.1284	0.1489

(b) LA-MAC

Mode / r_m	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.3356	0.1305	0.4801	0.0538	0.6644
2 *pps*	0.3358	0.1305	0.4800	0.0538	0.6642
10 *pps*	0.3358	0.1305	0.4799	0.0538	0.6642
20 *pps*	0.3364	0.1305	0.4794	0.0538	0.6636

(c) B-MAC

Table 3.8 – Scenario 4, dense star topology. Numerical details of time spent in each radio mode versus the traffic load. Network size is 6 nodes.

r_m \ Mode	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.7655	0.0079	0.0264	0.2002	0.2345
2 *pps*	0.4950	0.0083	0.0744	0.4223	0.5050
10 *pps*	0.4950	0.0083	0.0744	0.4223	0.5050
20 *pps*	0.4965	0.0083	0.0742	0.4211	0.5035

(a) X-MAC

r_m \ Mode	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.8850	0.0131	0.0184	0.0834	0.1150
2 *pps*	0.8837	0.0129	0.0215	0.0818	0.1162
10 *pps*	0.8436	0.0094	0.0148	0.1323	0.1564
20 *pps*	0.8351	0.0091	0.0157	0.1400	0.1649

(b) LA-MAC

r_m \ Mode	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 *pps*	0.3119	0.0996	0.5347	0.0538	0.6881
2 *pps*	0.3121	0.0996	0.5346	0.0537	0.6879
10 *pps*	0.3123	0.0996	0.5344	0.0537	0.6877
20 *pps*	0.3129	0.0996	0.5338	0.0537	0.6871

(c) B-MAC

Table 3.9 – Scenario 4, dense star topology. Numerical details of time spent in each radio mode versus the traffic load. Network size is 8 nodes.

3.6 Conclusions

Heterogeneous WSN providing support for multiple applications with sources with different characteristics, need adaptive MAC protocols able to support intense network dynamics.

We observe that existing MAC protocols are not suitable for simultaneously supporting traffic load variation and sources with heterogeneous traffic.

We also observe that relying on synchronized methods is very hard, because of the network scale and dynamic evolution. We propose in this chapter LA-MAC, a low-latency asynchronous access method for efficient forwarding in wireless sensor networks. In LA-MAC, receivers periodically organize the transmission of senders using a SCHEDULE message. Senders are allowed to transmit bursts of messages whose size depend on several factors such as priority of head of line message and the number of contenders. Thanks to SCHEDULE messages, LA-MAC efficiently supports message forwarding in large multi-hops networks resulting in low latency for high priority data and high delivery ratio.

We report on the results of extensive simulations that compare LA-MAC with B-MAC and X-MAC, two representative methods based on preamble sampling. We include the results for several important spatial scenarios that show excellent performance of LA-MAC with respect to latency, delivery ratio, and consumed energy.

Chapter 4

Energy Analysis of Preamble Sampling Based MAC

A major issue in wireless sensor networks is energy efficiency of nodes, their components and the protocols they use to interact with each other. Differently from other wireless networks such as cellular networks or IEEE 802.11, WSNs use batteries that must guarantee a lifetime of the order of years and that can be hardly recharged or replaced. Thus, the first step toward energy efficiency is the minimization of energy waste. MAC protocols play a crucial role in fighting energy waste in WSNs because they govern the use of radio equipment. If inefficiently used in fact, radio equipment may result in severe energy waste because listening to idle channel and overhearing of unnecessary radio frames exhaust scarce battery resource. This chapter presents a novel approach for modeling energy consumption of preamble sampling MAC protocols, partially deterministic and probabilistic. The attention is focused on Preamble Sampling MAC whose principles are random access of nodes and the use of preamble advertisement messages. The novel model is used to analyze and compare the consumption of three protocols : B-MAC [13], X-MAC [15], and LA-MAC. The results show higher energy efficiency of LA-MAC.

The chapter is organized as follows. After introducing in Sec. 4.1 the technical context, our motivations and contribution, Sec. 4.2 addresses the background, some definitions and assumptions. The analytic energy analysis is the subject of Sec. 4.3. We lastly propose a brief numerical validation of the model in Sec. 4.4 before to conclude the chapter in Sec. 4.5.

4.1 Introduction

Minimizing energy consumption is the main design goal of WSNs along with providing sufficient performance support for target applications. Medium Access Control methods play the key role in saving energy [7] because of the part taken by the radio in the overall energy budget. Therefore, designing an efficient access method consists in reducing the effects of energy waste. In random access MAC methods, in which wake-up schedules are neither organized nor synchronized, devices waste energy all the time when the radio equipment is unnecessarily switched on. In particular, main reasons of energy waste are both *idle listening* during which a device consumes energy while waiting for an eventual transmission and *overhearing* when it receives a frame sent to another device [7].

A viable way to save energy without neither synchronizing nor organizing schedules is by achieving low duty cycles : devices alternate long sleeping periods (radio switched off) and short active ones (radio switched on). The ideal situation would be that the receiver wakes up when possible transmissions of some devices start. However, node schedules are independent, thus, the challenge of MAC to achieve network-wide very low duty cycle is to minimize energy waste that results from un-synchronized wake-up schedules of nodes.

Nevertheless, achieving very low duty cycles without synchronization may affect system performance because of a possible long delay that a sender may experience before the intended receiver wakes up.

In addition to the un-synchronization problem, the lack of knowledge on instantaneous traffic load also influences network behavior, because it is related to the number of devices that want to access the channel and to the amount of time that devices need to empty their local buffer.

In complex, dense, and multi-hop networks, the distribution of instantaneous traffic load depends on traffic pattern, *e.g. convergecast*, *broadcast*, or *multicast*, traffic characteristics such as *packet generation rate* and several other factors such as the MAC method, the routing protocol, and density of nodes. As a consequence, traffic distribution is not uniformly spread over the network and it is difficult to predict.

Preamble sampling MACs save energy because they do not require explicit synchronization, however, they use long advertisement messages to *wake up* the receiver that can occupy the radio channel if instantaneous traffic load and node density are high. If instantaneous traffic load is high, then network is flooded with advertisement messages. The design of PS MAC must find a trade-off between energy efficiency and meeting application requirements such as low latency.

4.1.1 Motivations

The evaluation of energy consumption in complex wireless sensor networks that use preamble sampling MAC is difficult. Energy analyses published in the literature often base energy consumption of a given protocol upon the traffic generation rate of the network [17]. In our opinion, this approach does not fully reflect the complexity of the problem because even though all nodes generate periodic traffic with the same period, different network regions show different traffic loads. We propose to analyze the energy consumption with respect to the instantaneous network congestion in a given geographical area. This approach results more flexible, varying network congestion, different congestion situations can be modeled, reproducing situations that mimic areas with low density and traffic load as well as very crowded areas with high traffic load.

4.1.2 Contribution

In this chapter, we present a novel energy consumption analysis approach ; it estimates energy consumption of a WSN independently of traffic pattern and it is easy to apply to preamble sampling MAC methods. This chapter is based on a paper accepted for publication at the IEEE Personal Indoor Mobile Radio Communications conference [C3]. The analysis includes the cost of all radio operations involved in the transmission of data messages, namely the cost of transmitting, receiving, idle listening, overhearing and sleeping. The given analysis, is then used to analyze and compare B-MAC [13], X-MAC [15], and LA-MAC in terms of energy consumption (cf. Fig. 3.3). The novelty of the analysis lies in taking into account the relation of energy consumption and traffic load.

4.2 System Model

In complex, dense, and multi-hop networks, the distribution of instantaneous traffic load over the network is not uniform. For example, in the case of networks with *convergecast* traffic pattern (all messages go to one sink), traffic load is higher at nodes closer to the sink in terms of number of hops. Due to this effect, namely *funneling effect* [89], devices close to the sink exhaust their energy much faster than the others.

In our analysis, we are interested in analyzing energy consumption independently of traffic characteristics or traffic pattern ; to do so, we analyze the consumption of a localized area of a WSN, the extension to a large network is straightforward.

Devices are assumed deployed as a "star" network composed of a single receiving device, the *sink* or R_x and a group of N devices (T_{xi} with $i \in [1, ..., N]$) that may have data to send. All devices are within 1-hop radio coverage of each other, so each transmitted message that is broadcast in nature, is received by all nodes that leave their radio switched on. All N devices share a global message buffer for which B sets the number of queued messages ; B is then related to the instantaneous network congestion. Depending upon congestion degree, among all N devices, N_s of them have at least one packet to send ; those nodes plus the receiver are called *active* devices. Remaining devices have empty buffers and do not participate in the contention, nevertheless, they are prone to the *overhearing effect*. Thus, there are $N_o = N - N_s$ *over-hearers*. According to the global buffer state B, there are several combinations of how to distribute B packets among N devices ; hence, N_s and N_o vary as consequence. For instance, there can be B active devices each one with a single packet to send or less than B active devices with some of them having more than one buffered packet.

We explicitly separate the energy costs corresponding to transmission E_t, reception E_r, polling (listening for some radio activity in the channel) E_l, and sleeping E_s activities. E_o is the overall energy consumption of all over-hearers. The overall energy consumption E is the sum of all these energies and depends on a given MAC method (cf. Eq. 4.1).

$$E^{MAC}(B) = f(B, E_t, E_r, E_l, E_s, E_o, MAC) \tag{4.1}$$

According to the basics of preamble sampling approach, when the receiver wakes up, it polls the channel to detect some activity. Because of lack of explicit synchronization, it may happen that at the time when the receiver wakes up, the sender is already awaken and is performing channel polling too. The occurring probability of this event is $p = t_l/t_f$, with t_l and t_f, polling period and frame duration, respectively (cf. Sec. 3.3.1). If the receiver wakes up while the sender is polling the channel, it has the time to perform half of the polling process and then it listens for the first preamble of the sender. After the end of channel polling process, the sender starts sending the preamble series, thus, with probability (1-p) when the receiver wakes up, it receives a preamble immediately or in a very short time. In the remainder of this chapter, we say that with probability p transmitter and receiver are quasi-synchronized.

4.3 Energy Analysis

We focus on evaluating energy consumption of a network composed of N transmitters and one sink, the receiver. We provide a separated analytic evaluation of the energy consumption for three preamble sampling protocols : B-MAC, X-MAC, and LA-MAC.

We explicit the analytic expressions of energy consumption *E(B)* starting from the case of empty buffers B=*0* until the generalized expression for unknown values of B.

4.3.1 Empty Global Buffer (B=*0*)

If $B = 0$, all protocols behave in the same way : nodes periodically wake up, poll the channel for t_l seconds, then go back to sleep because of the absence of channel activity and messages to send. Overall network consumption is proportional to network population and only depends on the time that each node spends in polling and sleeping modes :

$$E^{ALL}(0) = (N + 1) \cdot (t_l \cdot P_l + t_s \cdot P_s) \tag{4.2}$$

4.3.2 Global Buffer with One Message (B=*1*)

If there is only one message to send, there are two active devices : the sender, that has a message in the buffer ($N_s = 1$) and the destination. Other devices ($N_o = N - 1$) have empty buffers, therefore, their energy consumption only depends on channel activity of active nodes that they can overhear and the amount of time that they spend in sleeping mode.

B-MAC ($B = 1$)

When message sender wakes up, it polls the channel for t_l seconds and then starts sending a long preamble that anticipates data transmission. Even if data are assumed unicast, the destination field is not included in preambles ; therefore, all neighbor nodes that progressively wake up need to hear both the preamble and the header of the following data to be able to know the identity of the intended destination. The cost for transmission is :

$$E_t^B(1) = (t_p^B + t_d) \cdot P_t \tag{4.3}$$

Devices are not synchronized and wake-up schedules are uniformly distributed across time, thus, each one hears an average time equal to the half duration of a long preamble before starting data reception. The cost of reception includes the cost of receiving the half duration of a long preamble added to the cost of receiving data. Energy consumption of each node depends upon probability of quasi-synchronization p :

$$E_r^B(1) = (p \cdot t_p^B + (1-p) \cdot \frac{t_p^B}{2} + t_d) \cdot P_r \tag{4.4}$$

The overall polling cost of current case involves both polling procedures of sender and receiver : the first one polls the channel for an entire polling period (t_l seconds) whereas the second one only for a duration that depends on p. The cost of polling activity is :

$$E_l^B(1) = (1 + \frac{p}{2}) \cdot t_l \cdot P_l \tag{4.5}$$

The cost of sleeping activity concerning the couple transmitter-receiver depends on the time that they do not spend in any mode among polling, receiving, or transmitting :

$$E_s^B(1) = (2 \cdot t_f - (\frac{t_p^B}{2} \cdot (p+3) + 2 \cdot t_d + t_l \cdot (1 + \frac{p}{2}))) \cdot P_s \tag{4.6}$$

With B-MAC, there is no difference in terms of energy consumption between overhearing and receiving a message. Therefore, the cost of overhearing activity is as follows :

$$E_o^B(1) = N_o \cdot (E_r^B(1) + p \cdot \frac{t_l}{2} \cdot P_l + (t_f - (p \cdot (\frac{t_l}{2} + t_p^B) + (1-p) \cdot \frac{t_p^B}{2} + t_d)) \cdot P_s) \tag{4.7}$$

X-MAC ($B = 1$)

When the sender wakes up, it polls the channel for t_l seconds and starts sending a series of unicast preambles separated by a gap for *early* ACK reception. Once the sink has received a short preamble, it clears it with an *early* ACK to stop the transmission of preambles and receive data. At this time the sender can transmit its message. After data reception, R_x remains in polling mode for an extra back-off time t_b that is used to receive other possible messages [15]. All devices that have no messages to send and that overhear channel activity go to sleep.

The expected number of preambles that are needed to *wake up* the receiver is γ^X :

$$\gamma^X = (\frac{t_l - t_a^X - t_p^X}{t_f})^{-1}, \tag{4.8}$$

where t_a^X is the duration of an *early* ACK message, and t_p^X the duration of a preamble message of the series. We remind that before the receiver wakes up and captures a preamble, there are $(\gamma^X - 1)$ preambles whose transmission energy is wasted. In X-MAC, the total amount of energy that results from the activity of transmitting one message depends on the average number of preambles that must be sent (γ^X) and the cost of *early* ACK reception. Provided that wake-up schedules of nodes are not synchronous, it may happen that when the receiver wakes up, the sender is already performing channel polling (transmitter and receiver are quasi-synchronized with probability p).

In the case of quasi-synchronization, the receiver stays an average duration equal to half of t_l in polling mode and then it is able to clear the very first preamble of the incoming series. With probability p, the cost of transmission only includes the cost of transmitting one preamble and the cost of receiving the *early* ACK that follows.

Otherwise, (with probability 1-p) the receiver wakes up after the end of the polling process of the sender ; thus, the receiver compels the sender to waste energy for the transmission of γ^X preambles and the wait for an *early* ACK (while waiting for *early* ACK, a node is in polling mode) before it can hear one preamble. Transmission cost is :

$$\begin{aligned} E_t^X(1) = &\ (1-p)\cdot\gamma^X\cdot t_p^X\cdot P_t + p\cdot t_p^X\cdot P_t + t_a^X\cdot P_r + t_d\cdot P_t \\ = &\ ((1-p)\cdot\gamma^X + p)\cdot t_p^X\cdot P_t + t_a^X\cdot P_r + t_d\cdot P_t \end{aligned} \tag{4.9}$$

The cost of receiving activity does not depend on p and it includes the transmission of one *early* ACK plus the reception of both data and preamble.

$$E_r^X(1) = (t_d + t_p^X)\cdot P_r + t_a^X\cdot P_t \tag{4.10}$$

With probability $1-p$ (no synchronization) the receiver wakes up while the sender is already transmitting a preamble (or it is waiting for an *early* ACK). Otherwise, (with probability p) the receiver stays in polling mode for an average duration of t_l.

If the active couple is quasi-synchronized, there is a period of time that both T_x and R_x simultaneously spend polling the channel, then, when the sender starts the transmission of the series of preambles, the receiver switches its radio to receiving mode. Within the whole channel polling cost for the sender, are included both the time spent polling the channel and the time that it waits for *early* ACK without any answer (event that happens $\gamma^X - 1$ times with probability $1-p$).

$$\begin{aligned} E_l^X(1) = &\ ((t_l + (1-p)\cdot(\gamma^X - 1)\cdot t_a^X) + ((1-p)\cdot\tfrac{t_p^X + t_a^X}{2} + p\cdot\tfrac{t_l}{2}) + t_b)\cdot P_l \\ = &\ ((1-p)\cdot(\tfrac{t_p^X + t_a^X}{2} + (\gamma^X - 1)\cdot t_a^X) + (\tfrac{p}{2} + 1)\cdot t_l + t_b)\cdot P_l \end{aligned} \tag{4.11}$$

The sleeping activity of the active couple is twice a frame duration less the time that both devices spend in one of the active modes :

$$\begin{aligned} E_s^X(1) = &\ (2\cdot t_f - (t_l + ((1-p)\cdot\gamma^X + p)\cdot(t_p^X + t_a^X) + t_d) - \\ &\ +(p\cdot\tfrac{t_l}{2} + t_p^X + t_a^X + (1-p)\cdot\tfrac{t_p^X + t_a^X}{2} + t_d + t_b))\cdot P_s \\ = &\ (2\cdot t_f - 2\cdot t_d - p\cdot\tfrac{t_l}{2} - t_p^X - t_a^X - (1-p)\cdot\tfrac{t_p^X + t_a^X}{2} - t_l)\cdot P_s + \\ &\ -(((1-p)\cdot\gamma^X + p)\cdot(t_p^X + t_a^X) - t_b)\cdot P_s \end{aligned} \tag{4.12}$$

In the same way as other devices, over-hearers can wake up at a random instant.

However, differently from active devices, as soon as they overhear some activity they immediately go back to sleep. Therefore, their energy consumption depends on the probability that such nodes wake up while the channel is busy or not. The probability that at the wake-up instant the channel is free, depends upon several factors such as polling duration, buffer states, and the number of senders. In Fig. 4.1, we show a tree containing all possible wake-up schedule combinations that may happen. In the tree, we consider as reference instant, the time at which the transmitter wakes up (root of the tree). With probability p, the transmitter (T_x) and the Receiver (R_x) are quasi-synchronized ; not synchronized (with probability $(1-p)$), otherwise. With probability $p\cdot p$ both the receiver and a generic over-hearer are quasi-synchronized with the transmitter, this is the Case 1 in the tree. In the remainder, we explicit the expressions for all possible combinations contained in the tree. Overall energy consumption resulting from the overhearing process is the sum of all combinations weighted by relative probabilities (cf. Eq. 4.22).

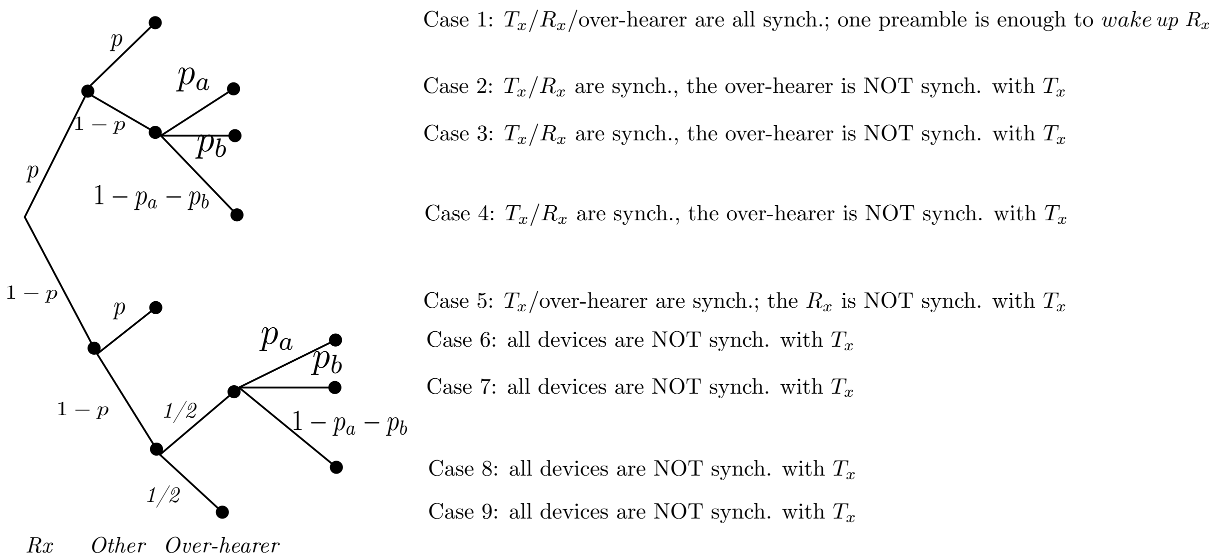


Figure 4.1 – Scenario with global buffer size B=1, X-MAC protocol. Tree containing all possible wake-up schedule combinations of T_x, R_x and over-hearers. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

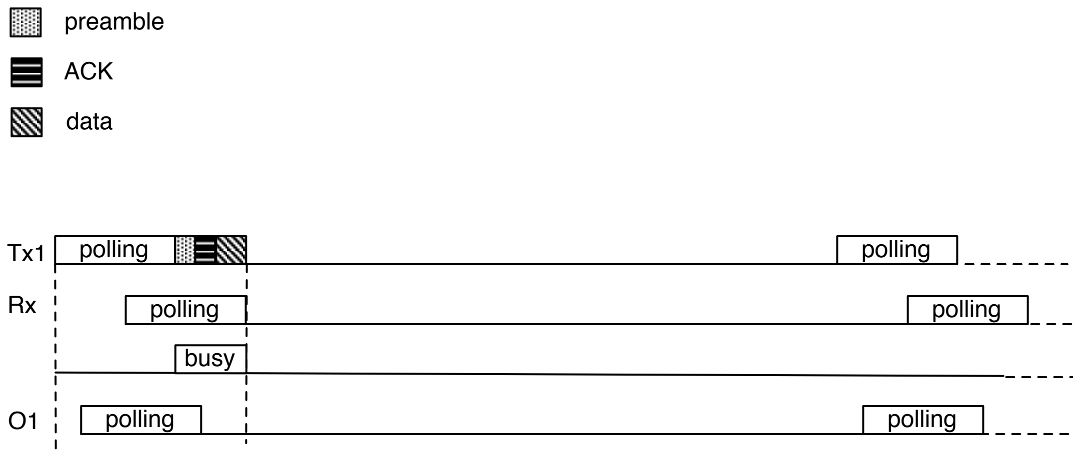


Figure 4.2 – X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Case 1.

- Case 1 : Sender, receiver and over-hearer are all quasi-synchronized (see Fig. 4.2). The over-hearer receives a preamble for the sink, then it goes back to sleep. Energy consumption of the overhearing action concerning Case 1 is :

$$E^X_{Case1,o} = \frac{t_l}{2} \cdot P_l + t^X_p \cdot P_r + (t_f - \frac{t_l}{2} - t^X_p) \cdot P_s \tag{4.13}$$

- Cases 2, 3, and 4 : The receiver is synchronized with the sender, whereas the over-hearer is not. When the over-hearer wakes up, it may overhear different messages such as a preamble (Case 2), an *early* ACK (Case 3), a data (Case 4) as well as a clear channel (Case 4 again). Possible situations are summarized in Fig. 4.3.

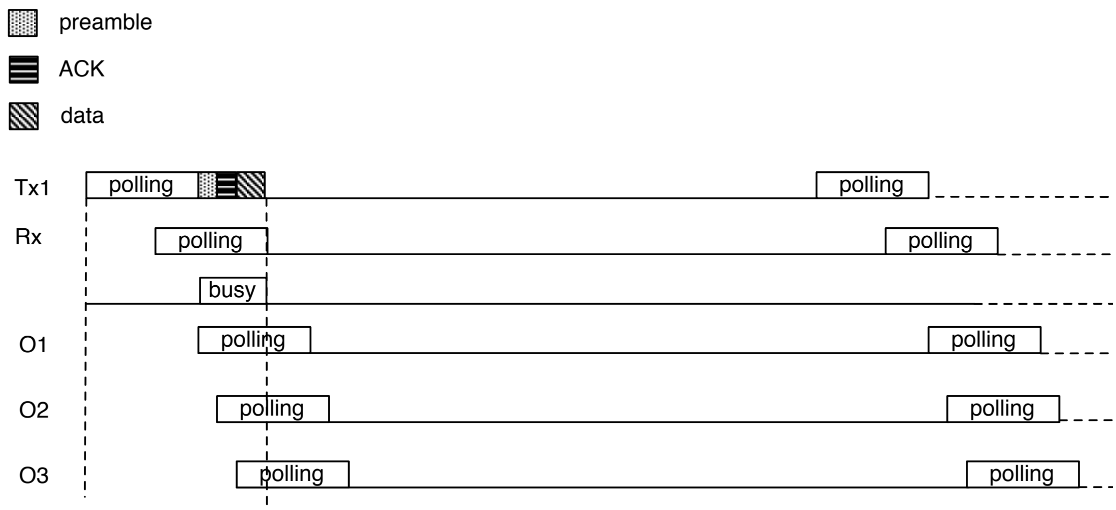


FIGURE 4.3 – X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 2, 3, and 4.

– Case 2 : If the over-hearer wakes up during a preamble transmission, it remains in polling mode without sensing any activity until the *early* ACK that follows the preamble is sent ; then, the over-hearer goes to sleep. The probability for the over-hearer to wake up during a preamble is $p_a = t_p^X / t_f$. Energy consumption resulting from Case 2 is as follows :

$$E^X_{Case_2,o} = \frac{t_p^X}{2} \cdot P_l + t_a^X \cdot P_r + (t_f - \frac{t_p^X}{2} - t_a^X) \cdot P_s \tag{4.14}$$

– Case 3 : If an over-hearer wakes up during an *early* ACK transmission, it stays in polling mode without detecting any channel activity until data is overheard ; afterwards it goes back to sleep. The probability for the over-hearer to wake up during an *early* ACK is $p_b = t_a^X / t_f$. Energy consumption concerning Case 3 is as follows :

$$E^X_{Case_3,o} = \frac{t_a^X}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_a^X}{2} - t_d) \cdot P_s \tag{4.15}$$

– Case 4 : The over-hearer either wakes up during data transmission or during the following silent period. In both events when the sender wakes up and senses the channel, it asserts that the channel is clear. From a consumption point of view these events are equivalent because if a message is already being transmitted by T_x when the over-hearer wakes up, it can not capture the begin of the transmission exactly like if there were not an ongoing transmission. Therefore, the over-hearer stays in polling mode for t_l seconds and goes back to sleep immediately after. The probability for this event to happen is $1 - p_a - p_b$. Energy consumption concerning

Case 4 is as follows :

$$E^X_{Case_4,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{4.16}$$

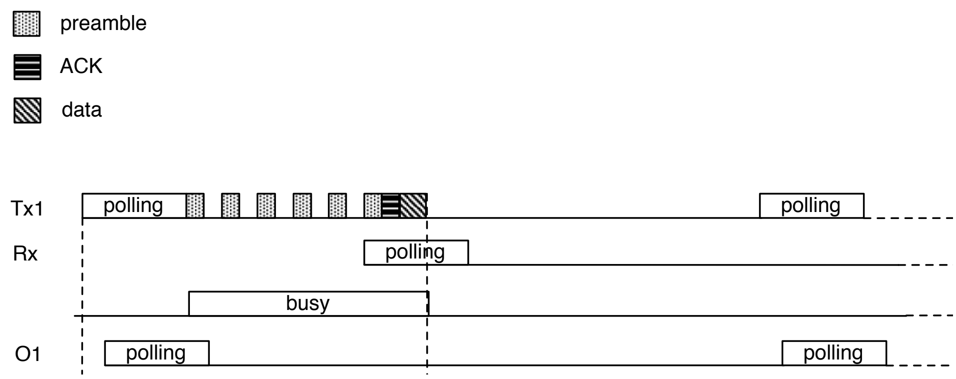


Figure 4.4 – X-MAC protocol, global buffer size $B = 1$. Overhearing situation for Case 5.

- Case 5 : Similarly to Case 1, if the over-hearer is quasi-synchronized with the transmitter, it overhears the first preamble even if the receiver is still sleeping ; then, it goes back to sleep. The energy cost is as follows :

$$E^X_{Case_5,o} = E^X_{Case_1,o} \tag{4.17}$$

- Cases 6, 7 and 8 : If neither the receiver nor the over-hearer are synchronized with the sender, it may happen that the receiver wakes up before the over-hearer (cf. Fig. 4.5). Therefore, similarly to Cases 2, 3 and 4, different situations are possible : Cases 6, 7, and 8 are similar to 2, 3, and 4, respectively. The costs are as follows :

$$E^X_{Case_6,o} = E^X_{Case_2,o} \tag{4.18}$$

$$E^X_{Case_7,o} = E^X_{Case_3,o} \tag{4.19}$$

$$E^X_{Case_8,o} = E^X_{Case_4,o} \tag{4.20}$$

- Case 9 : If the over-hearer wakes up before the R_x, as soon as it receives a preamble, it goes back to sleep. The cost concerning this Case is as follows :

$$E^X_{Case_9,o} = t^X_p \cdot P_r + \frac{t^X_p + t^X_a}{2} \cdot P_l + (t_f - \frac{t^X_p + t^X_a}{2} - t^X_p) \cdot P_s \tag{4.21}$$

The overall energy cost is the sum of all costs weighted by the probability of the given case to happen :

$$E^X_o(1) = N_o \cdot \sum_{i=1}^{9} p_{Case_i} \cdot E^X_{Case_i,o} \tag{4.22}$$

LA-MAC ($B = 1$)

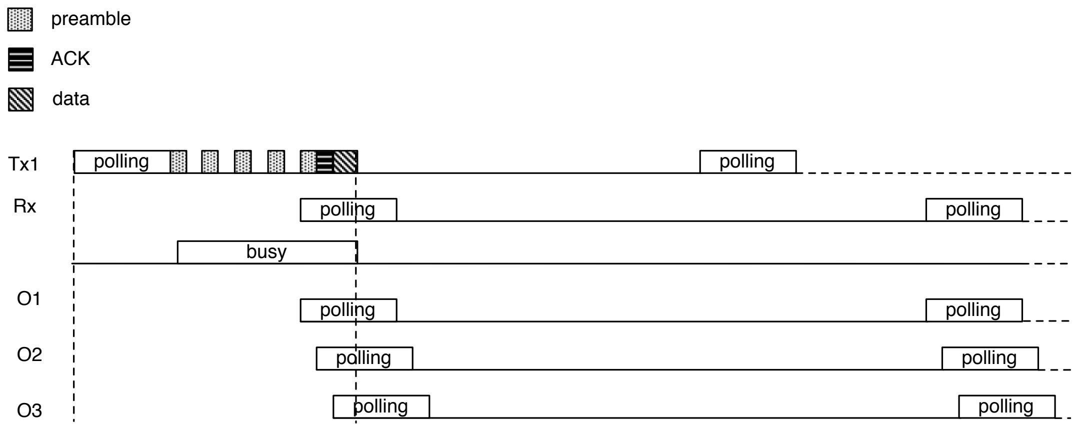


FIGURE 4.5 – X-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 6, 7, and 8.

In LA-MAC, when the unique sender wakes up, it polls the channel for t_l seconds and then it transmits a series of preambles as in X-MAC. However, differently from X-MAC after *early* ACK reception, the sender goes back to sleep and waits for SCHEDULE message to be sent. Moreover, when the receiver captures one preamble, it clears it with an *early* ACK and completes its polling period in order to detect additional possible preambles to clear. Immediately after the end of polling period, the receiver processes the requests that has cleared and broadcasts a SCHEDULE message containing a local and temporal transmission organization. In LA-MAC, over-hearers go to sleep as soon as they receive any unicast message (preamble, *early* ACK or data) as well as a SCHEDULE (that is a broadcast message).

Because of the lack of synchronization, the expected number of preambles needed to *wake up* the receiver follows X-MAC expression with different sizes of preambles (t_p^L) and *early* ACKs (t_a^L) (cf. Eq. 3.1). The cost of transmission activity concerning the current case ($E_t^L(1)$) is similar to the cost of X-MAC excepting for the cost of SCHEDULE message that must be added :

$$\begin{aligned} E_t^L(1) = & \quad (1-p)\cdot\gamma^L\cdot t_p^L\cdot P_t + p\cdot t_p^L\cdot P_t + t_a^L\cdot P_r + t_d\cdot P_t + t_g\cdot P_r \\ = & \quad ((1-p)\cdot\gamma^L + p)\cdot t_p^L\cdot P_t + (t_a^L + t_g)\cdot P_r + t_d\cdot P_t \end{aligned} \tag{4.23}$$

The cost of reception activity depends on the duration of preamble, *early* ACK, data and SCHEDULE messages :

$$E_r^L(1) = (t_p^L + t_d)\cdot P_r + (t_a^L + t_g)\cdot P_t \tag{4.24}$$

Differently from X-MAC, the receiver completes its polling period independently of the number of cleared preambles, so, its radio remains in polling mode for the duration of t_l seconds less the time spent in receiving a preamble and transmitting *early* ACK. The cost of polling activity is as follows :

$$E_l^L(1) = ((t_l + (1-p)\cdot(\gamma^L - 1)\cdot t_a^L) + (t_l - t_p^L - t_a^L))\cdot P_l \tag{4.25}$$

When the active nodes are not transmitting, receiving or polling the channel they can sleep. Cost of sleeping activity is as follows :

$$E_s^L(1) = (2\cdot t_f - (t_l + (1-p)\cdot\gamma^L\cdot t_p^L + p\cdot t_p^L + t_a^L + (1-p)\cdot(\gamma^L - 1)\cdot t_a^L + t_d + t_g) - (t_l + t_d + t_g))\cdot P_s \tag{4.26}$$

As in X-MAC as soon as over-hearers receive a message they go back to sleep. Therefore, their energy consumption depends on the probability that such nodes wake up while the channel is busy or clear. All possible combinations of wake-up schedules with relative probabilities are shown in the tree depicted in Fig. 4.6. Overall energy cost is the sum of all costs weighted by the probability of the given case to happen (cf. Eq. 4.38).

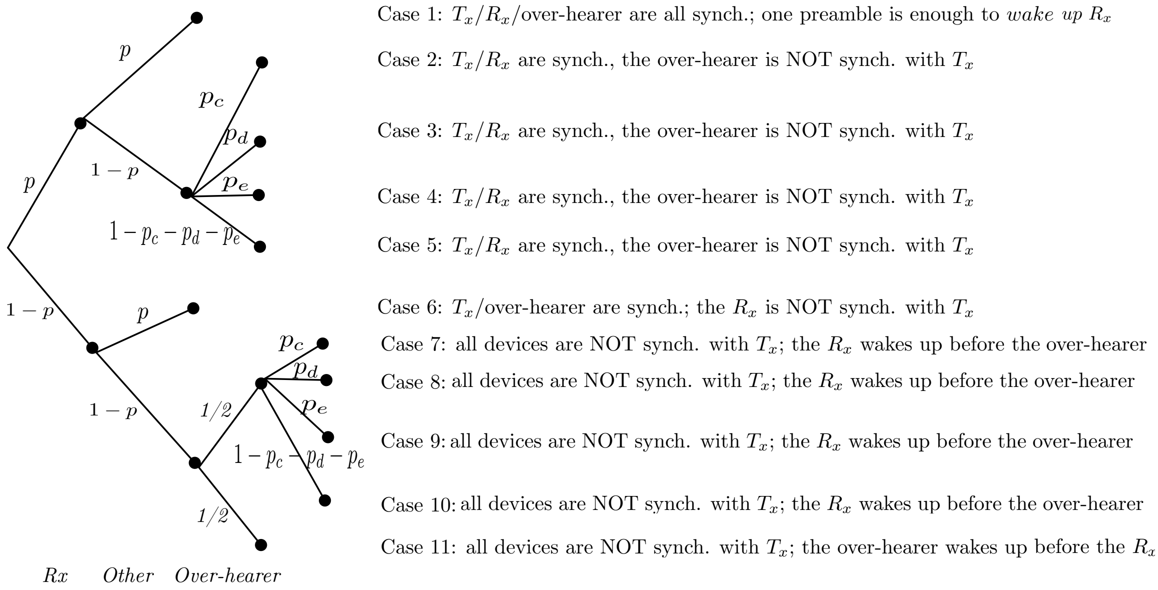


FIGURE 4.6 – Scenario with global buffer size B=1, LA-MAC protocol. Tree containing all possible wake-up schedule combinations of T_x, R_x and over-hearers. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

- Case 1 : Sender, receiver, and over-hearer are quasi-synchronized. The over-hearer captures a preamble for the sink and goes back to sleep (cf. Sec. 4.7). The probability of this event to occur is $p \cdot p$. Energy cost concerning Case 1 is as follows :

$$E^L_{Case1,o} = \frac{t_l}{2}\cdot P_l + t_p^L \cdot P_r + (t_f - \frac{t_l}{2} - t_p^L)\cdot P_s \tag{4.27}$$

- Cases 2, 3, 4 and 5 : The receiver is synchronized with the sender whereas the over-hearer is not. Thus, when the over-hearer wakes up it may receive different messages : a preamble (Case 2), an *early* ACK (Case 3), a SCHEDULE (Case 4) or data (Case 5) as well as clear channel (Case 5 again) (see Fig. 4.8).
  - Case 2 : If an over-hearer wakes up during a preamble transmission, it stays in polling mode without receiving any message until it overhears the following *early* ACK. Afterwards it goes back to sleep. Probability of this event to occur is $p \cdot$

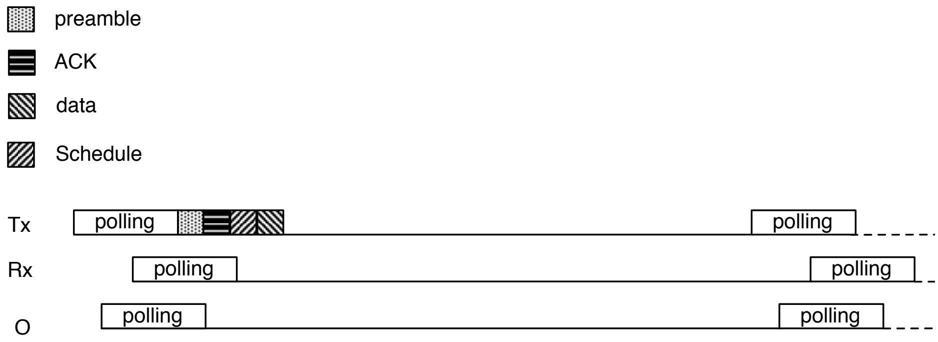


FIGURE 4.7 – LA-MAC protocol, global buffer size $B = 1$. Overhearing situation for Case 1.

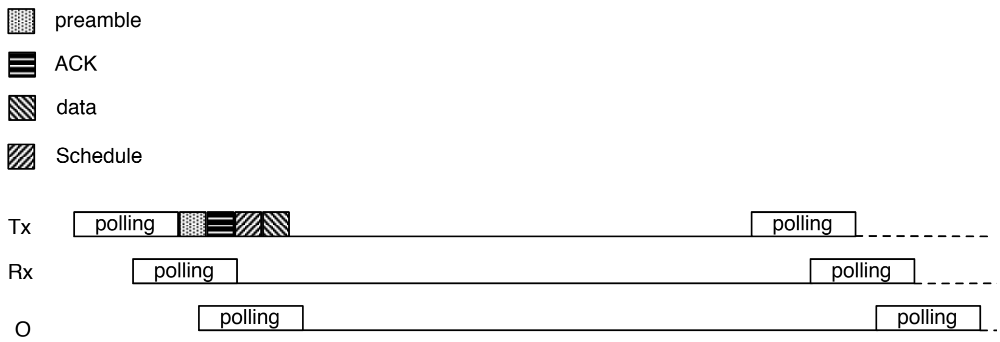


FIGURE 4.8 – LA-MAC protocol, global buffer size $B = 1$. Overhearing situations for Cases 2, 3, 4 and 5.

$(1-p) \cdot p_c$, where $p_c = t_p^L/t_f$ represents the event that wake-up instant of the over-hearer happens immediately after the end of polling process of the sender. The energy cost concerning Case 2 is as follows :

$$E^L_{Case_2,o} = \frac{t_p^L}{2} \cdot P_l + t_a^L \cdot P_r + (t_f - \frac{t_p^L}{2} - t_a^L) \cdot P_s \tag{4.28}$$

– Case 3 : If the over-hearer wakes up during an *early* ACK transmission, it senses a silent period and overhears the following SCHEDULE message. Afterwards, it goes back to sleep. The probability of this event to occur is $p \cdot (1-p) \cdot p_d$, where $p_d = t_a^L/t_f$ includes the event that wake-up instant of the over-hearer happens after the transmission of a preamble. p_d neglects the time that elapses between the end of the *early* ACK and the end of channel polling process of the receiver. In other words, with p_d we assume that SCHEDULE message is sent immediately after the transmission of *early* ACK. The energy cost concerning Case 3 is as follows :

$$E^L_{Case_3,o} = \frac{t_a^L}{2} \cdot P_l + t_g \cdot P_r + (t_f - \frac{t_a^L}{2} - t_g) \cdot P_s \tag{4.29}$$

– Case 4 : If the over-hearer wakes up during the transmission of a SCHEDULE message, it does not sense any channel activity and remains in polling mode until it receives a data, then, it goes to sleep. The probability of this event to occur is $p \cdot (1-p) \cdot p_e$, with $p_e = t_g/t_f$ we assume that the wake-up instant of the over-hearer happens in average at the middle of SCHEDULE transmission. The energy cost concerning Case 4 is as follows :

$$E^L_{Case_4,o} = \frac{t_g}{2} \cdot P_l + t_d \cdot P_r + (t_f - \frac{t_g}{2} - t_d) \cdot P_s \tag{4.30}$$

– Case 5 : The over-hearer either wakes up during data transmission or senses a free channel because both sender and receiver are already sleeping. Therefore, the over-hearer polls the channel for t_l seconds and goes back to sleep. The probability of this event to happen is $p \cdot (1-p) \cdot (1 - p_c - p_d - p_e)$. The energy cost concerning Case 5 is the following :

$$E^L_{Case_5,o} = t_l \cdot P_l + (t_f - t_l) \cdot P_s \tag{4.31}$$

– Case 6 : Similarly to Case 1, if the over-hearer is quasi-synchronized with the sender, with probability $(1-p) \cdot p$ the energy cost is as follows :

$$E^L_{Case_6,o} = \frac{t_l}{2} \cdot P_l + t^L_p \cdot P_r + (t_f - \frac{t_l}{2} - t^L_p) \cdot P_s \tag{4.32}$$

– Cases 7, 8, 9, and 10 : If neither the receiver nor the over-hearer are synchronized with sender, it may happen that the receiver wakes up before the over-hearer. We distinguish the situations of quasi-synchronization of the couple over-hearer-receiver and lack of synchronization as shown in Fig. 4.9.

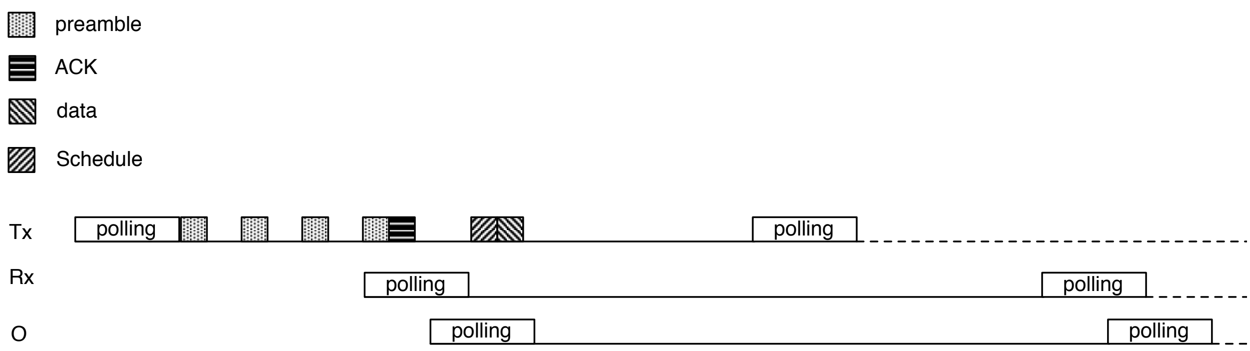


Figure 4.9 – LA-MAC. Possible wake-up instants of over-hearers. Cases 7, 8, 9 and 10.

In Cases 7 and 8, the over-hearer is quasi-synchronized with the receiver :

– Case 7 : There is a probability to overhear a preamble. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_c$. Consumption in this case is the same as in Case 2 :

$$E^L_{Case_7,o} = E^L_{Case_2,o} \tag{4.33}$$

– Case 8 : There is a probability to overhear an *early* ACK. Such a probability is equal to $(1-p) \cdot (1-p) \cdot 1/2 \cdot p_d$. Consumption in this case is the same as in Case 3 :

$$E^L_{Case_8,o} = E^L_{Case_3,o} \tag{4.34}$$

If the over-hearer and the receiver are not synchronized among each other :

- Case 9 : There is a probability to overhear a SCHEDULE. Such a probability is equal to $(1-p)\cdot(1-p)\cdot 1/2\cdot p_e$. Consumption in this case is the same as in Case 4 :

$$E^L_{Case_9,o} = E^L_{Case_4,o} \tag{4.35}$$

- Case 10 : There is a probability to overhear a data message. Such a probability is equal to $(1-p)\cdot(1-p)\cdot 1/2\cdot(1-p_c-p_d-p_e)$. Consumption in this case is the same as in Case 5 :

$$E^L_{Case_{10},o} = E^L_{Case_5,o} \tag{4.36}$$

- Case 11 : Otherwise, if the over-hearer wakes up before the destination, it receives one preamble (whichever preamble among γ^L) and goes back to sleep. The cost in this case is as follows :

$$E^L_{Case_{11},o} = \frac{t^L_p + t^L_a}{2}\cdot P_l + t^L_p\cdot P_r + (t_f - \frac{t^L_p + t^L_a}{2} - t^L_p)\cdot P_s \tag{4.37}$$

The overall energy cost is the sum of all the elementary costs weighted by the probability of the given case to happen :

$$E^L_o(1) = N_o\cdot\sum_{i=1}^{11} p_{Case_i}\cdot E^L_{Case_i,o}. \tag{4.38}$$

4.3.3 Global Buffer with Two Messages (B=2)

If $B=2$, there can be either one sender with two buffered messages, or two senders (T_{x1} and T_{x2}) with only one buffered message each. The number of over-hearers is $N_o = N-1$ if there is just one sender, $N_o = N-2$, otherwise. The probability that two messages are in different buffers is equal to $(N-1)/N$, where N is the number of nodes excluding the sink.

B-MAC ($B=2$)

Overall energy consumption for transmission and reception when $B\geq 1$ is linear with the global number of packets in the buffer, independently of how packets are distributed across the different local buffers, *i.e.,* independently of the number of senders. In fact, because of the long preamble to send ($t^b_p = t_f$), there can be only one sender per frame. Thus, the following relation exists : $E^B(B) = B\cdot E^B(1) = B\cdot(E^B_t(1)+E^B_r(1)+E^B_l(1)+E^B_s(1)+E^B_o(1))$.

Such a relation highlights the limitations of B-MAC protocol, since high-loaded traffic can hardly been addressed resulting in both high latency and energy consumption.

X-MAC ($B=2$)

After the reception of a data message, the receiver remains in polling mode for an extra back-off time t_b during which it can possibly receive a second message. The energy consumed for the transmission of the first packet is the same as the energy defined in the previous section ($E^X_t(1)$) ; then, an additional cost for the transmission of the second message must be considered.

With probability $1/N$, both packets are in the same buffer ; two different senders are involved, otherwise. Differently from B-MAC, the distribution of messages in the buffers impacts protocol behavior.

In case of multiple senders, the overall energy consumption depends on the way how wake-up instants of the active devices are scheduled with respect to each others. For example, assume that device A_1 with a message to send wakes up and receives a preamble from another sender A_2 ; thus, A_1 must remain in receiving mode and postpones its transmission until the intended receiver clears the preambles of A_2 with an *early* ACK, then A_1 sends its data message during the extra back-off time. In this example, the whole time that A_1 spends overhearing preambles from A_2 is wasted.

All the combinations of wake-up schedules that involve receiver, senders and overhearers are summarized in the tree depicted in Fig. 4.10. Wake-up instants of different devices are all independent and we assume that each protocol frame begins at the wake-up instant of T_{x1}.

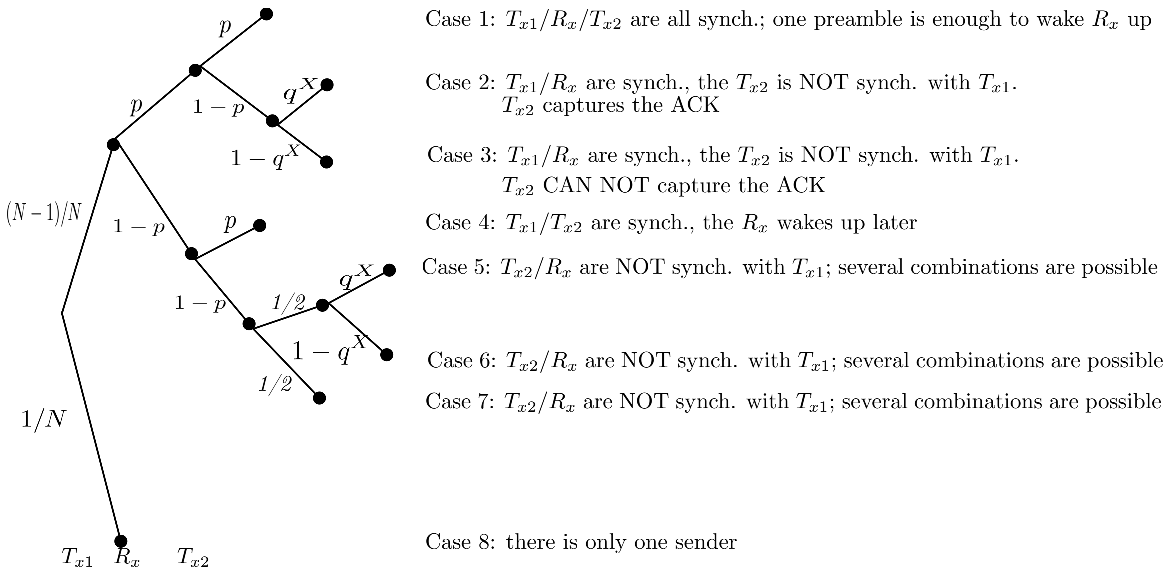


FIGURE 4.10 – Scenario with global buffer size B=2, X-MAC protocol. Tree containing all possible wake-up schedule combinations of T_{x1}, T_{x2} and R_x. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

– Case 1 : There are two senders and one receiver, all quasi-synchronized. The very first preamble sent by T_{x1} is cleared by the receiver that sends an *early* ACK ; T_{x2} hears both the preamble and the *early* ACK. The probability of this scenario to occur is $p_{Case_1} = (N-1)/N \cdot p \cdot p$. The costs are as follows :

$$E^X_{Case_1,t}(2) = t^X_p \cdot P_t + t^X_a \cdot P_r + (t^X_p + t^X_a) \cdot P_r + 2 \cdot t_d \cdot P_t \tag{4.39}$$

$$E^X_{Case_1,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \tag{4.40}$$

$$E^X_{Case1,l}(2) = (t_l + \frac{t_l}{2} + \frac{t_l}{2}) \cdot P_l \tag{4.41}$$

$$E^X_{Case1,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + t_d) - (\frac{t_l}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \tag{4.42}$$

Depending on wake-up instants of the over-hearers, several situations may happen. If an over-hearer is quasi-synchronized with one of the three active devices (the receiver or one of the two senders), it senses a busy channel (cf. Fig. 4.11). In this case, each over-hearer that polls the channel for some time may overhear a preamble, an *early* ACK or a data. For simplicity, we consider the worst case, *i.e.,* we assume that the over-hearer polls the channel for an average duration equal to half of a polling period and then it overhears a data (*i.e.,* the longest message that can be overheard). The probability to wake up during a busy period is $p^X_{case1,B=2} = (t^X_p + t^X_a + 2 \cdot t_d)/t_f$. Otherwise, if the over-hearer wakes up while channel is free, it polls the channel for t_l seconds and then goes back to sleep. The energy cost is as follows :

$$\begin{aligned} E^X_{Case1,o}(2) = \; & N_o \cdot (p^X_{case1,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s)) + \\ + \; & N_o \cdot ((1 - p^X_{case1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \end{aligned} \tag{4.43}$$

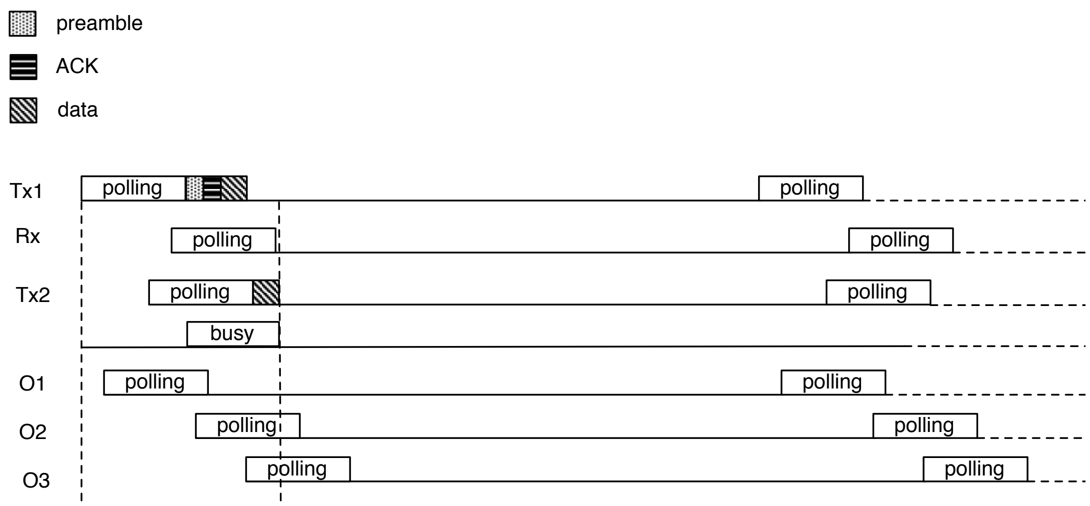


FIGURE 4.11 – X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 1.

– Case 2 : The first sender and receiver are quasi-synchronized, whereas T_{x2} is not synchronized with T_{x1} (cf. Fig. 4.12). The only possibility for the second sender to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver. This event happens with probability $q^X = (t_l - t^X_a)/t_f$. The probability of this scenario is $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^X$.

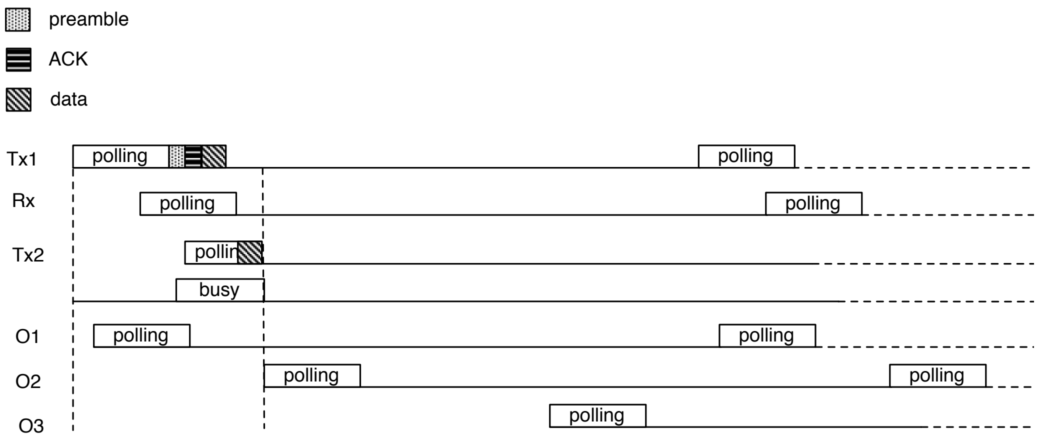


FIGURE 4.12 – X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 2.

Energy consumption concerning Case 2 is about the same as Case 1, with different event probability. Energy consumption of different activities becomes :

$$E^X_{Case_2,t}(2) = E^X_{Case_1,t}(2) - t^X_p \cdot P_r \tag{4.44}$$

$$E^X_{Case_2,r}(2) = E^X_{Case_1,r}(2) \tag{4.45}$$

$$E^X_{Case_2,l}(2) = E^X_{Case_1,l}(2) - \frac{t_l - t^X_p}{2} \cdot P_l \tag{4.46}$$

$$E^X_{Case_2,s}(2) = E^X_{Case_1,s}(2) + \frac{t_l + t^X_p}{2} \cdot P_l \tag{4.47}$$

We assume that the probability of busy channel is the same as in Case 1. So, overhearing consumption is unchanged :

$$E^X_{Case_2,o}(2) = E^X_{Case_1,o}(2) \tag{4.48}$$

– Case 3 : With probability $1 - q^X$, the second sender wakes up too late and cannot capture the *early* ACK. If this happens, it goes back to sleep and it transmits its data during the next frame. The energy cost is the sum of the transmission cost of the first packet in the current frame and the second packet in the following frame. The cost of second frame is the same as $E^X(1)$. This scenario happens with probability $p_{Case_3} = (N-1)/N \cdot p \cdot (1-p) \cdot (1-q^X)$. The energy costs in this case are the following :

$$E^X_{Case_3,t}(2) = t^X_p \cdot P_t + t^X_a \cdot P_r + t_d \cdot P_t + E^X_t(1) \tag{4.49}$$

$$E^X_{Case_3,r}(2) = t^X_p \cdot P_r + t^X_a \cdot P_t + t_d \cdot P_r + E^X_r(1) \tag{4.50}$$

$$E^X_{Case_3,l}(2) = (t_l + t_l + \frac{t_l}{2}) \cdot P_l + E^X_l(1) \tag{4.51}$$

$$E^X_{Case_3,s}(2) = (3 \cdot t_f - (t_l + t^X_p + t^X_a + t_d) - t_l - (\frac{t_l}{2} + t^X_p + t^X_a + t_d)) \cdot P_s + E^X_s(1) \quad (4.52)$$

In the second frame, the local buffer of the first sender is empty, thus, it can be counted as an over-hearer. Therefore, number of over-hearers does not change from first to second frame. The energy cost per over-hearer is the same as in the case of a single message to send ($B = 1$), that is :

$$E^X_{Case_3,o}(2) = (N_o + (N_o + 1)) \cdot \frac{E^X_o(1)}{N_o + 1} \quad (4.53)$$

– Case 4 : The first and second senders are quasi-synchronized whereas the receiver wakes up later. If this happens, the first sender sends a series of preambles until the receiver wakes up and sends an *early* ACK ; second sender hears the entire series of preambles and then sends its data during the extra back-off time (cf. Fig. 4.13). Between short preambles, both senders poll channel waiting for an *early* ACK from receiver. The probability of this scenario to happen is $p_{Case_4} = (N-1)/N \cdot (1-p) \cdot p$. The energy costs in this case are as follows :

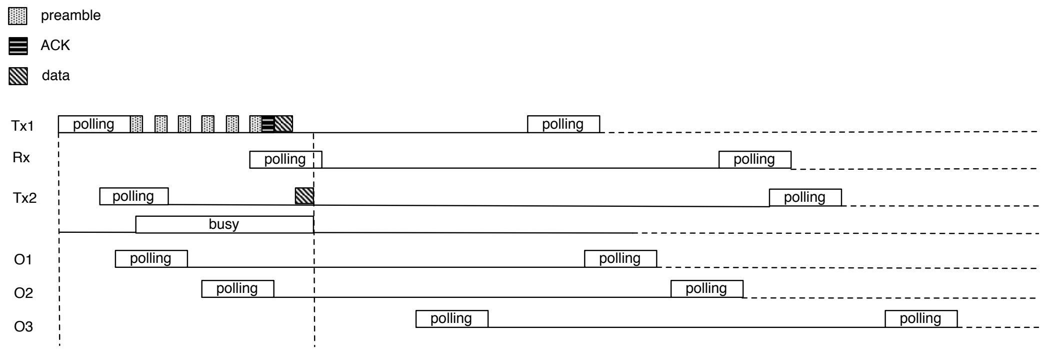


FIGURE 4.13 – X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 4.

$$E^X_{Case_4,t}(2) = \gamma^X \cdot t^X_p \cdot (P_t + P_r) + 2 \cdot t^X_a \cdot P_r + 2 \cdot t_d \cdot P_t \quad (4.54)$$

$$E^X_{Case_4,r}(2) = (t^X_p + 2 \cdot t_d) \cdot P_r + t^X_a \cdot P_t \quad (4.55)$$

$$E^X_{Case_4,l}(2) = (t_l + \frac{t_l}{2} + 2 \cdot (\gamma^X - 1) \cdot t^X_a + \frac{t^X_p + t^X_a}{2}) \cdot P_l \quad (4.56)$$

$$\begin{aligned} E^X_{Case_4,s}(2) = \quad & (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) - (\tfrac{t_l}{2} + \gamma^X \cdot (t^X_p + t^X_a) + t_d) \\ - \quad & (\tfrac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \end{aligned} \quad (4.57)$$

Because the receiver wakes up after both senders, the probability that an over-hearer wakes up during a transmission of a preamble is higher than in previous cases. If this happens, the over-hearer stays in polling mode for a very short time, it overhears a message (most likely a preamble) and then it goes back to sleep. For simplicity we assume that the over-hearer polls the channel for a duration equal to half of $(t^X_p + t^X_a)$ and then overhears an entire preamble. The probability of busy channel is thus $p^X_{case4,B=2} = (\gamma^X \cdot (t_p^X + t_a^X) + 2 \cdot t_d)/t_f$. The overhearing cost in this case is the following :

$$\begin{aligned} E^X_{Case_4,o}(2) = \quad & N_o \cdot (p^X_{case4,B=2} \cdot (\frac{t_p^X+t_a^X}{2} \cdot P_l + t_p^X \cdot P_r + (t_f - \frac{t_p^X+t_a^X}{2} - t_p^X) \cdot P_s) \\ + \quad & (1 - p^X_{case4,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s)) \end{aligned} \tag{4.58}$$

- Cases 5, 6, and 7 : The second sender and receiver are not synchronized with first sender ; behavior of the protocol depends on which device among T_{x2} and R_x wakes up first.
  - Case 5 : The receiver wakes up first as illustrated in Fig. 4.14. Similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver. This event happens with probability $q^X = (t_l - t_a^X)/t_f$. However, there is also the possibility for T_{x2} to capture a preamble sent by T_{x1}. Such eventuality can happen with probability $u^X = (t_p^X + t_a^X)/(2 \cdot t_p^X + t_a^X)$. This scenario happens with probability $p_{Case_5} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot q^X$. The energy costs become :

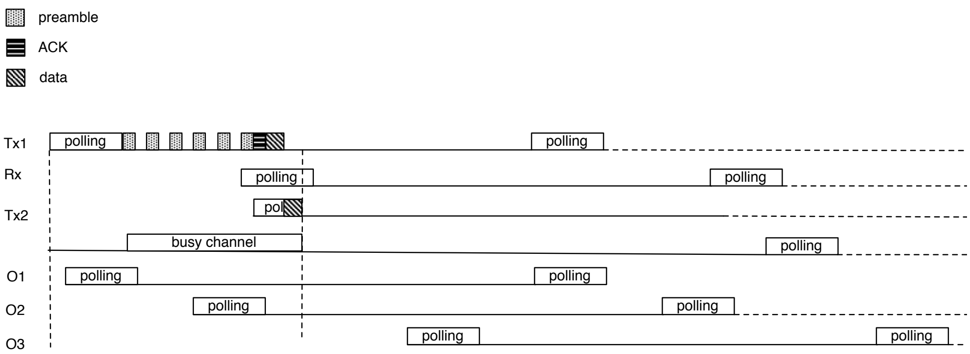


Figure 4.14 – X-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 5.

$$E^X_{Case_5,t}(2) = (\gamma^X \cdot t_p^X + t_d) \cdot P_t + t_a^X \cdot P_r + (u^X \cdot t_p^X + t_a^X) \cdot P_r + t_d \cdot P_t \tag{4.59}$$

$$E^X_{Case_5,r}(2) = (t_p^X + 2 \cdot t_d) \cdot P_r + t_a^X \cdot P_t \tag{4.60}$$

$$E^X_{Case_5,l}(2) = (t_l + (\gamma^X - 1) \cdot t_a^X + \frac{t_p^X + t_a^X}{2} + u^X \cdot \frac{t_p^X + t_a^X}{2} + (1 - u^X) \cdot \frac{t_p^X}{2}) \cdot P_l \tag{4.61}$$

$$\begin{aligned} E^X_{Case_5,s}(2) = \quad & (3 \cdot t_f - (t_l + \gamma^X \cdot (t_p^X + t_a^X) + t_d) \\ - \quad & (u^X \cdot \frac{t_p^X+t_a^X}{2} + (1 - u^X) \cdot \frac{t_p^X}{2} + u^X \cdot t_p^X + t_a^X + t_d) \\ - \quad & (\frac{t_p^X+t_a^X}{2} + t_p^X + t_a^X + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{4.62}$$

As in Case 4, the over-hearer senses a busy channel because of the transmission of preambles ; so when it wakes up we assume that if spends half of $(t_p^X + t_a^X)$ time in polling mode before overhearing an entire preamble. The probability of busy channel when the over-hearer wakes up is $p^X_{case5} = p^X_{case4}$. The overhearing costs are as follows :

$$E^X_{Case_5,o}(2) = E^X_{Case_4,o}(2) \tag{4.63}$$

– Case 6 : The receiver wakes up first. Similarly to Case 3, with probability $(1-q^X)$, T_{x2} wakes up too late and cannot capture the *early* ACK from the receiver. Thus, it goes back to sleep and transmits its data during the next frame. This scenario happens with probability $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2} \cdot (1-q^X)$. The energy consumption is as follows :

$$E^X_{Case_6,t}(2) = \gamma^X \cdot t^X_p \cdot P_t + t^X_a \cdot P_r + t_d \cdot P_t + E^X_t(1) \tag{4.64}$$

$$E^X_{Case_6,r}(2) = (t^X_p + t_d) \cdot P_r + t^X_a \cdot P_t + E^X_r(1) \tag{4.65}$$

$$E^X_{Case_6,l}(2) = (t_l + (\gamma - 1) \cdot t^X_a) \cdot P_l + t_l \cdot P_l + \frac{t^X_p + t^X_a}{2} \cdot P_l + E^X_l(1) \tag{4.66}$$

$$E^X_{Case_3,s}(2) = (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) + t_l + (\frac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + t_d)) \cdot P_s + E^X_s(1) \tag{4.67}$$

$$E^X_{Case_6,o}(2) = E^X_{Case_3,o}(2) = 2 \cdot E^X_o(1) \tag{4.68}$$

– Case 7 : The second transmitter wakes up first, it over-hears a part of the series of preambles until the receiver wakes up and sends an *early* ACK.
On the average, when T_{x2} wakes up, it polls the channel for a duration that is equal to the half of the gap between two successive short preambles : $(t^X_p + t^X_a)/2$. After that, it over-hears an average number of $\lfloor \gamma^X/2 \rfloor$ short preambles before the receiver wakes up and stops the series of preambles by sending an *early* ACK. The probability of this scenario is $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot \frac{1}{2}$. The energy costs become :

$$E^X_{Case_7,t}(2) = (\gamma^X \cdot t^X_p + t_d) \cdot P_t + t^X_a \cdot P_r + (\lfloor \frac{\gamma^X}{2} \rfloor \cdot t^X_p + t^X_a) \cdot P_r + t_d \cdot P_t \tag{4.69}$$

$$E^X_{Case_7,r}(2) = (t^X_p + t_d) \cdot P_r + t^X_a \cdot P_t + t_d \cdot P_r \tag{4.70}$$

$$E^X_{Case_7,l}(2) = (t_l + (\gamma^X - 1) \cdot t^X_a) \cdot P_l + ((\lfloor \frac{\gamma^X}{2} \rfloor - 1) \cdot t^X_a + \frac{t^X_p + t^X_a}{2}) \cdot P_l + \frac{t^X_p + t^X_a}{2} \cdot P_l \tag{4.71}$$

$$\begin{aligned} E^X_{Case_7,s}(2) = \;& (3 \cdot t_f - (t_l + \gamma^X \cdot (t^X_p + t^X_a) + t_d) - (\tfrac{t^X_p + t^X_a}{2} + \lfloor \tfrac{\gamma^X}{2} \rfloor \cdot (t^X_p + t^X_a) + t_d) \\ - \;& (\tfrac{t^X_p + t^X_a}{2} + t^X_p + t^X_a + 2 \cdot t_d)) \cdot P_s \end{aligned} \tag{4.72}$$

From the over-hearers point of view, this case is equivalent to Cases 4 and 5. The consumption is :

$$E^X_{Case_7,o}(2) = E^X_{Case_4,o}(2) \tag{4.73}$$

– Case 8 : There is only one sender and it has two messages in its buffer. This last scenario happens with a probability equal to $p_{Case_8} = 1/N$. The costs are :

$$E^X_{Case_8,t}(2) = E^X_t(1) + t_d \cdot P_t \tag{4.74}$$

$$E^X_{Case_8,r}(2) = E^X_r(1) + t_d \cdot P_r \tag{4.75}$$

$$E^X_{Case_8,l}(2) = E^X_l(1) - t_d \cdot P_l \tag{4.76}$$

$$E^X_{Case_8,s}(2) = E^X_s(1) - t_d \cdot P_s \tag{4.77}$$

When the sender is unique, energy consumption of the over-hearers can be assumed about the same as the one in case of a global buffer with one packet to send ($B = 1$). We have :

$$E^X_{Case_8,o}(2) = E^X_o(1) \tag{4.78}$$

The overall energy cost is the sum of all costs of each scenario, weighted by the probability of the scenario to happen (as showed in Fig. 4.10) :

$$E^X(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E^X_{Case_i}(2) \tag{4.79}$$

LA-MAC ($B = 2$)

Energy consumption $E^L(2)$ depends on the number of senders as well as on how wake-up instants occur. All different combinations of wake-up instants with their probabilities are given in the tree illustrated in Fig. 4.15. With the probability equal to $(N-1)/N$, there are two senders, a single sender otherwise. Cases 1-7 refer to situations in which two senders are involved, whereas Case 8 refers to a scenario with one sender. We introduce now some probabilities that are used in the remainder of this section. As previously defined, let $p = t_l/t_f$ be the probability of quasi-synchronization between two devices. The probability that T_{x2} polls the channel and over-hears the *early* ACK from R_x is $q^L = (t_l - t^L_a)/t_f$. The probability that R_x receives a preamble from T_{x2} before the end of its polling period is $w^L = (t_l - 2 \cdot t^L_p - t^L_a)/t_f$.

If none of the previous situations happen, R_x is not able to send an *early* ACK to T_{x2}. In this case, the address of T_{x2} is not included in the SCHEDULE message and it must wait until next frame to send data.

- Case 1 : The three active devices are all quasi-synchronized. The first preamble is instantly cleared by the receiver ; T_{x2} hears both the preamble and the *early* ACK. This event happens with probability $p_{Case_1} = (N-1)/N \cdot p \cdot p$.
  Depending whether the second transmitter succeeds or not in sending in time a preamble (before the end of polling period of the receiver, *i.e.*, with probability w^L), one or two frames are needed for sending two data messages. The energy costs are as follows :

$$\begin{aligned} E^L_{Case_1,t}(2) = &\ (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ + &\ w^L \cdot (t^L_p \cdot (P_r + P_t) + 2 \cdot t^L_a \cdot P_r + t_g \cdot P_r + t_d \cdot P_t) \\ + &\ (1 - w^L) \cdot (t^L_p \cdot P_r + t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \tag{4.80}$$

$$E^L_{Case_1,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w_L \cdot (t^L_p \cdot P_r + t^L_a \cdot P_t + t_d \cdot P_r) + (1 - w^L) \cdot E^L_r(1) \tag{4.81}$$

$$E^L_{Case_1,l}(2) = (2 \cdot t_l - t^L_p - t^l_a) \cdot P_l + w^L \cdot (-(t^L_p + t^L_a) + \frac{t_l}{2}) \cdot P_l + (1 - w_l) \cdot (\frac{t_l}{2} \cdot P_l + E^L_l(1)) \tag{4.82}$$

$$\begin{aligned} E^L_{Case_1,s}(2) = &\ (2 \cdot t_f - (t_l + t^L_p + t^L_a + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\ w^L \cdot (-t_d + t_f - (\tfrac{t_l}{2} + 2 \cdot (t^L_p + t^L_a) + t_g + t_d)) \cdot P_s \\ + &\ (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t^L_p + t^L_a)) \cdot P_s + E^L_s(1)) \end{aligned} \tag{4.83}$$

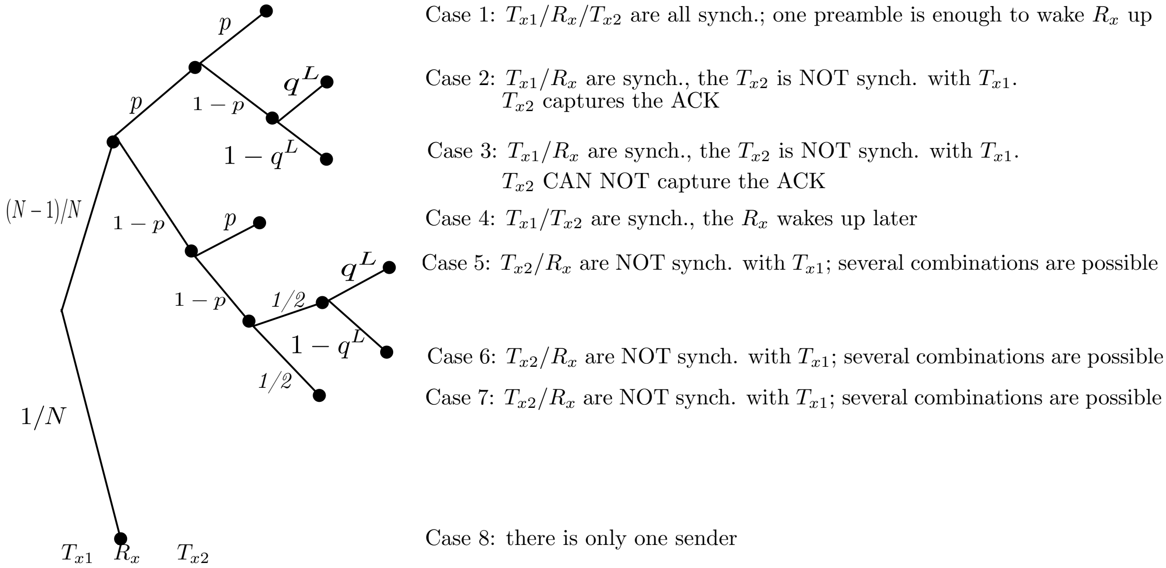


FIGURE 4.15 – Scenario with global buffer size B=2, LA-MAC protocol. Tree containing all possible wake-up schedule combinations of T_{x1}, T_{x2} and R_x. Branches are independent, thus, the probability at leaf is the product of probabilities of the whole path from the root to the leaf.

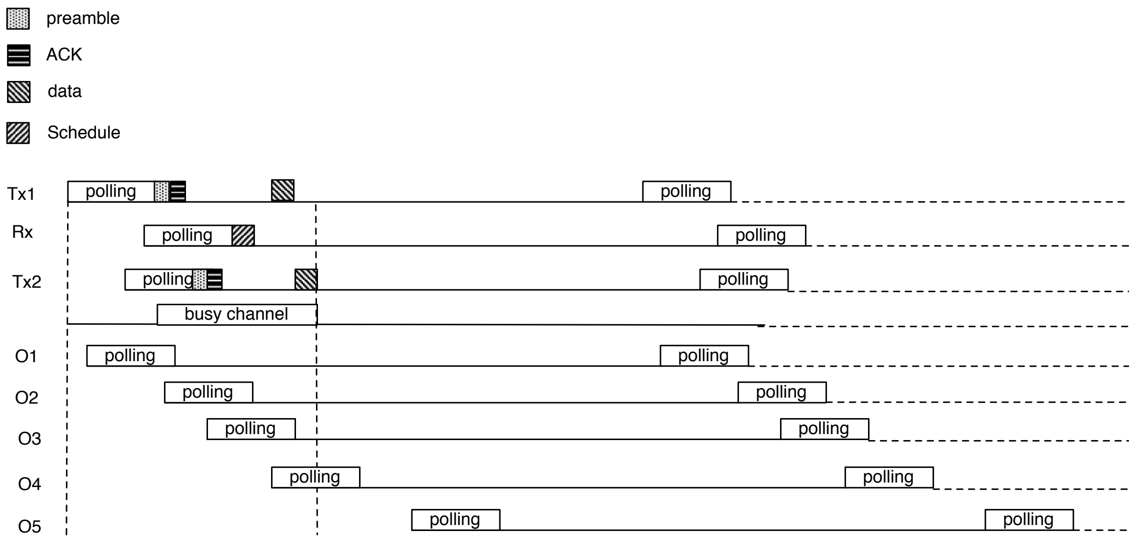


FIGURE 4.16 – LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 1.

As far as over-hearers are concerned, several situations may happen depending on their instants of wake-up. If an over-hearer is quasi-synchronized with one of the three active devices (T_{x1}, T_{x2} or R_x), it senses a busy channel (cf. Fig. 4.16). When an over-hearer wakes up, it polls the channel for some time and then it can overhear a message (that can be a preamble, an *early* ACK, a SCHEDULE or a data). We consider the worst case, *i.e.,* we assume that the over-hearer polls the channel for an average time equal to half the duration of t_l and then it overhears a data (the longest message that can be sent). The probability to wake up during a busy period is $p^L_{case1.1,B=2} = (2 \cdot (t^L_p + t^L_a + t_d) + t_g)/t_f$ if two data are sent within the same frame, $p^L_{case1.2,B=2} = (t^L_p + t^L_a + t_d + t_g)/t_f$, otherwise. If the over-hearer wakes up while channel is free, it polls the channel for t_l seconds, then it goes to sleep. The overhearing cost is the following :

$$\begin{aligned} E^L_{Case_1,o}(2) = &\ N_o \cdot w^L \cdot p^L_{case1.1,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + &\ N_o \cdot w^L \cdot (1 - p^L_{case1.1,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + &\ N_o \cdot (1 - w^L) \cdot p^L_{case1.2,B=2} \cdot (\tfrac{t_l}{2} \cdot P_l + t_d \cdot P_r + (t_f - \tfrac{t_l}{2} - t_d) \cdot P_s) \\ + &\ N_o \cdot (1 - w^L) \cdot (1 - p^L_{case1.2,B=2}) \cdot (t_l \cdot P_l + (t_f - t_l) \cdot P_s) \\ + &\ (1 - w^L) \cdot E^L_o(1) \end{aligned} \tag{4.84}$$

– Case 2 : The first transmitter and the receiver are quasi-synchronized whereas T_{x2} is not synchronized with T_{x1}. R_x first clears a preamble of T_{x1}, and then waits in polling mode for another possible preamble to come until the end of its polling period. Immediately after the end of polling period, the receiver broadcasts a SCHEDULE message. The only possibility for T_{x2} to be included in the SCHEDULE of the current frame is to send a preamble before that the receiver stops polling the channel and that R_x sends it an *early* ACK ; this event happens with probability $q^L = (t_l - t^L_a)/t_f$. If T_{x2} sends a preamble too late, it may happen that there is not enough remaining time for the receiver to receive a preamble and send an *early* ACK (probability of this event is w^L) before the end of its polling period. Case 2 happens with probability $p_{Case_2} = (N-1)/N \cdot p \cdot (1-p) \cdot q^L$. The energy costs in the current case are as follows :

$$\begin{aligned} E^L_{Case_2,t}(2) = &\ (t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r \\ + &\ w^L \cdot ((t^L_p + t_d) \cdot P_t + (2 \cdot t^L_a + t_g) \cdot P_r) \\ + &\ (1 - w^L) \cdot (t^L_a \cdot P_r + E^L_t(1)) \end{aligned} \tag{4.85}$$

$$E^L_{Case_2,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + w^L \cdot ((t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t) + (1 - w^L) \cdot E^L_r(1) \tag{4.86}$$

$$E^L_{Case_2,l}(2) = (2 \cdot t_l - t^L_p - t^L_a) \cdot P_l + w^L \cdot (-(t^L_p + t^L_a) + \frac{t^L_p}{2}) \cdot P_l + (1 - w^L) \cdot (\frac{t_l}{2} \cdot P_l + E^L_l(1)) \tag{4.87}$$

$$\begin{aligned} E^L_{Case_2,s}(2) = &\ (2 \cdot t_f - (t_l + t^L_p + t^L_a + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ + &\ w_l \cdot (-t_d + t_f - (\tfrac{t^L_p}{2} + t^L_p + 2 \cdot t^L_a + t_g + t_d)) \cdot P_s \\ + &\ (1 - w^L) \cdot ((t_f - (\tfrac{t_l}{2} + t^L_a)) \cdot P_s + E^L_s(1)) \end{aligned} \tag{4.88}$$

We assume that the probability of busy channel is the same as in Case 1. So, consumption is assumed to be the same, that is :

$$E^L_{Case_2,o}(2) = E^L_{Case_1,o}(2) \tag{4.89}$$

– Case 3 : With probability $(1 - q^L)$, T_{x2} wakes up too late and cannot capture the acknowledge sent by the receiver to T_{x1}. In this case, the second sender goes back to sleep and transmits its data during the next frame. Nevertheless, depending on its exact wake-up instant, T_{x2} can spend more or less time in each radio mode. Let us define the remaining time $t_{remain} = (t_f - t_l/2 - t_p^L - t_a^L)$ as being the part of the receiver frame during which the second sender can wake up. Let us also define a variable that behaves like a test of positivity : $test = max(t_l/2 - t_p^L - t_a^L, 0)$; such $test$ variable states that "T_{x2} wakes up in the time that follows the transmission of *early* ACK by R_x". Case 3 happens with probability $p_{Case_3} = (N-1)/N \cdot p \cdot (1-p) \cdot (1-q^L)$. The energy costs are the following :

$$E^L_{Case_3,t}(2) = (t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + E_t^L(1) \tag{4.90}$$

$$E^L_{Case_3,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + E_r^L(1) + \frac{test}{t_{remain}} \cdot t_g \cdot P_r + \frac{t_g}{t_{remain}} \cdot t_d \cdot P_r \tag{4.91}$$

$$\begin{aligned} E^L_{Case_3,l}(2) = {} & (2 \cdot t_l - t_p^L - t_a^L) \cdot P_l + E_l^L(1) \\ + {} & \tfrac{test}{t_{remain}} \cdot \tfrac{test}{2} \cdot P_l + \tfrac{t_g}{t_{remain}} \cdot \tfrac{t_g}{2} \cdot P_l + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot t_l \cdot P_l \end{aligned} \tag{4.92}$$

$$\begin{aligned} E^L_{Case_3,s}(2) = {} & (2 \cdot t_f - (t_l + t_p^L + t_a^L + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s + E_s^L(1) \\ + {} & \tfrac{test}{t_{remain}} \cdot (t_f - t_g) \cdot P_s + \tfrac{t_g}{t_{remain}} \cdot (t_f - t_d) \cdot P_s + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot (t_f - t_l) \cdot P_s \end{aligned} \tag{4.93}$$

Since there are two frames for sending two data messages, the energy spent by overhearers is about the same as the one detailed in previous section ($B = 1$), that is :

$$E^L_{Case_3,o}(2) = \frac{N_o + N_o + 1}{N_o - 1} \cdot E_o^L(1) \tag{4.94}$$

– Case 4 : The first and second senders are quasi-synchronized, whereas the receiver wakes up later (see Fig. 4.17). In this case, T_{x1} sends a series of preambles until the receiver wakes up and sends the *early* ACK. Even if the second sender overhears some preambles, it must remain awake until *early* ACK is sent. This scenario happens with probability $p_{Case_4} = (N-1)/N \cdot (1-p) \cdot p$ and the resulting energy costs are as follows :

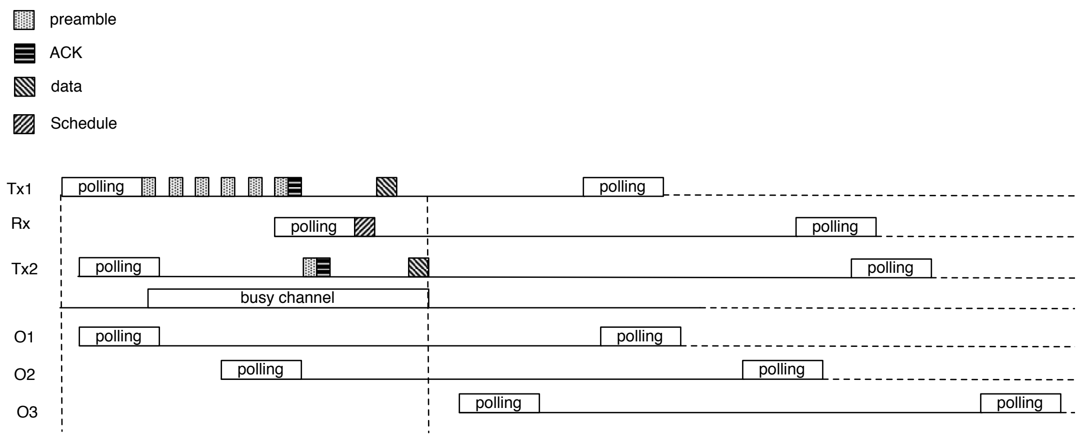


FIGURE 4.17 – LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 4.

$$E^L_{Case_4,t}(2) = (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + (t_p^L + t_d) \cdot P_t + (\gamma^L \cdot t_p^L + 2 \cdot t_a^L + t_g) \cdot P_r \tag{4.95}$$

$$E^L_{Case_4,r}(2) = (t^L_p + t_d)\cdot P_r + (t^L_a + t_g)\cdot P_t + (t^L_p + t_d)\cdot P_r + t^L_a \cdot P_t \quad (4.96)$$

$$E^L_{Case_4,l}(2) = (t_l + (\gamma^L - 1)\cdot t^L_a + t_l - t^L_p - t^L_a)\cdot P_l + (-(t^L_p + t^L_a) + \frac{t_l}{2} + (\gamma^L - 1)\cdot t^L_a)\cdot P_l \quad (4.97)$$

$$\begin{aligned} E^L_{Case_4,s}(2) = & \ (2\cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d))\cdot P_s \\ + & \ (-t_d + t_f - \tfrac{t_l}{2} - (\gamma^L + 1)\cdot(t^L_p + t^L_a) - t_g - t_d)\cdot P_s \end{aligned} \quad (4.98)$$

If the receiver wakes up after the couple of senders, the probability that an over-hearer wakes up during a transmission of a preamble is high. If this happens, the over-hearer stays in polling mode for a very short time, overhears a message (most likely a preamble) and then goes back to sleep. We consider the *pessimistic* case where the over-hearer polls the channel for a duration equal to $\frac{t_l}{2}$ and then overhears the longest possible type of message, *i.e.,* a data. The probability of busy channel when the over-hearer wakes up is $p^L_{case4} = ((\gamma^L + 1)\cdot(t^L_p + t^L_a) + t_g + 2\cdot t_d)/t_f$. The overhearing cost is as follows :

$$E^L_{Case_4,o}(2) = N_o\cdot(p^L_{case4}\cdot(\frac{t_l}{2}\cdot P_l + t_d\cdot P_r + (t_f - \frac{t_l}{2} - d)\cdot P_s) + (1 - p^L_{case4})\cdot(t_l\cdot P_l + (t_f - t_l)\cdot P_s)) \quad (4.99)$$

– Cases 5, 6, and 7 : According to these three cases, T_{x2} and R_x are not synchronized with T_{x1} ; the behavior of the protocol depends on which device wakes up first among the second transmitter and the receiver.
  – Case 5 : R_x wakes up first ; similarly to Case 2, the only possibility for the second transmitter to send data in the current frame is to poll the channel and capture the *early* ACK of the receiver (cf. Fig. 4.18). This event happens with probability $q^L = (t_l - t^L_a)/t_f$. As previously explained, the energy spent for the transmission of the second data message depends on the probability of the receiver to capture in time the preamble sent by the second sender. This fifth scenario has occurring probability given by $p_{Case_5} = (N-1)/N\cdot(1-p)\cdot(1-p)\cdot 1/2\cdot q^L$. The energy cost concerning this case is as follows :

$$\begin{aligned} E^L_{Case_5,t}(2) = & \ (\gamma^L \cdot t^L_p + t_d)\cdot P_t + (t^L_a + t_g)\cdot P_r \\ + & \ w^L\cdot((t^L_p + t_d)\cdot P_t + (2\cdot t^L_a + t_g)\cdot P_r) \\ + & \ (1 - w^L)\cdot(t^L_a\cdot P_r + E^L_t(1)) \end{aligned} \quad (4.100)$$

$$E^L_{Case_5,r}(2) = (t^L_p + t_d)\cdot P_r + (t^L_a + t_g)\cdot P_t + w^L\cdot((t^L_p + t_d)\cdot P_r + t^L_a\cdot P_t) + (1 - w^L)\cdot E^L_r(1) \quad (4.101)$$

$$\begin{aligned} E^L_{Case_5,l}(2) = & \ (t_l + (\gamma^L + 1)\cdot t^L_a + t_l - (t^L_p + t^L_a))\cdot P_l \\ + & \ w^L\cdot(-(t^L_p + t^L_a) + \tfrac{t^L_p}{2})\cdot P_l \\ + & \ (1 - w^L)\cdot(\tfrac{t_l}{2}\cdot P_l + E^L_l(1)) \end{aligned} \quad (4.102)$$

$$\begin{aligned} E^L_{Case_5,s}(2) = & \ (2\cdot t_f - (t_l + \gamma^L\cdot(t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d))\cdot P_s \\ + & \ w_l\cdot(-t_d + t_f - (\tfrac{t^L_p}{2} + t^L_p + 2\cdot t^L_a + t_g + t_d))\cdot P_s \\ + & \ (1 - w^L)\cdot((t_f - (\tfrac{t_l}{2} + t^L_a))\cdot P_s + E^L_s(1)) \end{aligned} \quad (4.103)$$

As in the previous scenario, the over-hearer senses a very busy channel because of the transmission of preambles ; it wakes up, stays half of t_l in polling mode and than overhears a data. The overhearing cost concerning this case is the following :

$$E^L_{Case_5,o}(2) = E^L_{Case_4,o}(2) \quad (4.104)$$

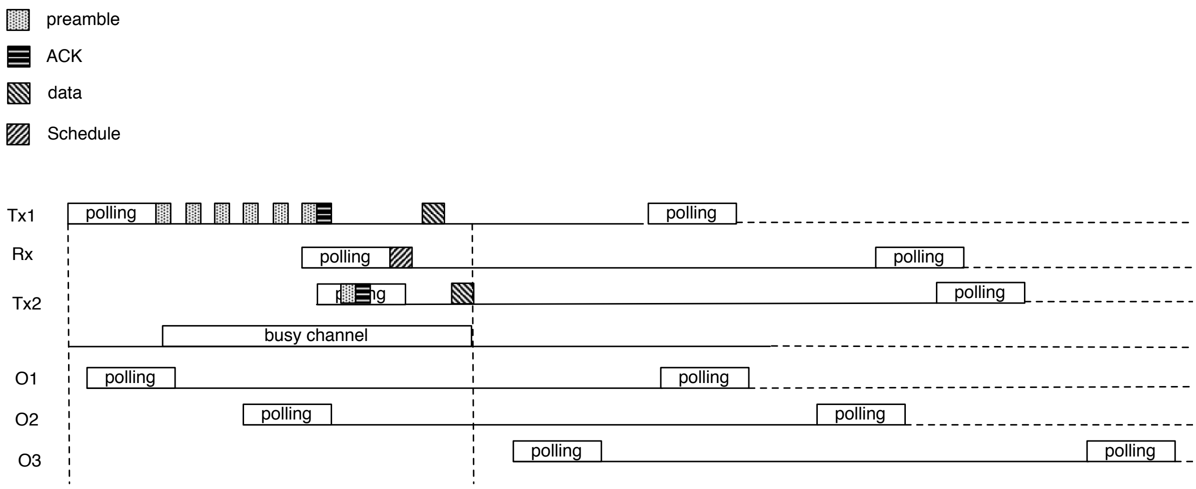


FIGURE 4.18 – LA-MAC protocol, global buffer size $B = 2$. Overhearing situations for Case 5.

– Case 6 : The receiver wakes up before T_{x2}, similarly to Case 3. With probability $(1-q^L)$, the second sender wakes up too late and cannot capture the *early* acknowledge. In this case, it goes back to sleep and transmits its data during the next frame. The first sender needs to send a series of preambles to wake up the receiver. The probability of this scenario to happen is given by $p_{Case_6} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2 \cdot (1-q^L)$.
We now provide the expressions for t_{remain} and $test$ variables. We have :

$$\begin{aligned} t_{remain} = &\quad t_f - \tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L \\ test = &\quad max(\tfrac{t_p^L + t_a^L}{2} - t_p^L - t_a^L, 0) \end{aligned} \tag{4.105}$$

The energy costs of Case 6 are the following :

$$E^L_{Case_6,t}(2) = (\gamma^L \cdot t_p^L + t_d) \cdot P_t + (t_a^L + t_g) \cdot P_r + E_t^L(1) \tag{4.106}$$

$$E^L_{Case_6,r}(2) = (t_p^L + t_d) \cdot P_r + (t_a^L + t_g) \cdot P_t + E_r^L(1) + \frac{test}{t_{remain}} \cdot t_g \cdot P_r + \frac{t_g}{t_{remain}} \cdot t_d \cdot P_r \tag{4.107}$$

$$\begin{aligned} E^L_{Case_6,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t_a^L + t_l - t_p^L - t_a^L) \cdot P_l + E_l^L(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot \tfrac{test}{2} \cdot P_l + \tfrac{t_g}{t_{remain}} \cdot \tfrac{t_g}{2} \cdot P_l + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot t_l \cdot P_l \end{aligned} \tag{4.108}$$

$$\begin{aligned} E^L_{Case_6,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t_p^L + t_a^L) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s + E_s^L(1) \\ + &\quad \tfrac{test}{t_{remain}} \cdot (t_f - t_g) \cdot P_s + \tfrac{t_g}{t_{remain}} \cdot (t_f - t_d) \cdot P_s + (1 - \tfrac{test + t_g}{t_{remain}}) \cdot (t_f - t_l) \cdot P_s \end{aligned} \tag{4.109}$$

From the over-hearers point of view, this scenario is comparable to the one of Case 3, resulting in the following cost equal to :

$$E^L_{Case_6,o}(2) = E^L_{Case_3,o}(2) \tag{4.110}$$

– Case 7 : T_{x2} wakes up before R_x, so, it is ready to send a preamble immediately after the transmission of the *early* ACK destined to T_{x1}. The second transmitter hears a part of the strobed preamble of the first transmitter : in average, it hears

$\lfloor \gamma^L/2 \rfloor$ preambles. This scenario has an occurring probability equal to $p_{Case_7} = (N-1)/N \cdot (1-p) \cdot (1-p) \cdot 1/2$. The energy costs are as follows :

$$E^L_{Case_7,t}(2) = (\gamma^L \cdot t^L_p + t_d) \cdot P_t + (t^L_a + t_g) \cdot P_r + (\lfloor \frac{\gamma^L}{2} \rfloor \cdot t^L_p + 2 \cdot t^L_a t_g) \cdot P_r + (t^L_p + t_d) \cdot P_t \tag{4.111}$$

$$E^L_{Case_7,r}(2) = (t^L_p + t_d) \cdot P_r + (t^L_a + t_g) \cdot P_t + (t^L_p + t_d) \cdot P_r + t^L_a \cdot P_t \tag{4.112}$$

$$\begin{aligned} E^L_{Case_7,l}(2) = &\quad (t_l + (\gamma^L - 1) \cdot t^L_a + t_l - t^L_p - t^L_a) \cdot P_l \\ &+ \quad (-(t^L_p + t^L_a) + \tfrac{t^L_p + t^L_a}{2} + (\lfloor \tfrac{\gamma^L}{2} \rfloor - 1) \cdot t^L_a) \cdot P_l \end{aligned} \tag{4.113}$$

$$\begin{aligned} E^L_{Case_7,s}(2) = &\quad (2 \cdot t_f - (t_l + \gamma^L \cdot (t^L_p + t^L_a) + t_g + t_d) - (t_l + t_g + t_d)) \cdot P_s \\ &+ \quad (-t_d + t_f - \tfrac{t^L_p + t^L_a}{2} - (\lfloor \tfrac{\gamma^L}{2} \rfloor + 1) \cdot (t^L_p + t^L_a) - t_g - t_d) \cdot P_s \end{aligned} \tag{4.114}$$

From the over-hearers point of view, this case is equivalent to Case 4. The cost is as follows :

$$E^L_{Case_7,o}(2) = E^L_{Case_4,o}(2) \tag{4.115}$$

– Case 8 : There is only one sender that sends two messages in a row. This last scenario happens with probability $p_{Case_8} = 1/N$. The resulting costs are as follows :

$$E^L_{Case_8,t}(2) = E^L_t(1) + t_d \cdot P_t \tag{4.116}$$

$$E^L_{Case_8,r}(2) = E^L_r(1) + t_d \cdot P_r \tag{4.117}$$

$$E^L_{Case_8,l}(2) = E^L_l(1) \tag{4.118}$$

$$E^L_{Case_8,s}(2) = E^L_s(1) - 2 \cdot t_d \cdot P_s \tag{4.119}$$

When the sender is unique, overhearing consumption can be assumed the same as the case of $B = 1$. The cost in this case becomes :

$$E^L_{Case_8,o}(2) = E^L_o(1) \tag{4.120}$$

The overall energy cost is the sum of all energy consumption of each case weighted by the probability of the case to happen (as showed on the Fig. 4.15) :

$$E^L(2) = \sum_{i=1}^{8} p_{Case_i} \cdot E^L_{Case_i} \tag{4.121}$$

4.3.4 Global Buffer with More Than Two Messages (B>2)

We now derive the generic expression of energy consumption for larger values of B. Following the same approach of the previous cases would be complex and tedious because of the large number of possible wake-up combinations, thus, to provide the generalized expression we follow a different approach based on the maximum number of packets that can be sent during a single frame.

B-MAC ($B > 2$)

With B-MAC protocol, if the global buffer state is larger than 1, energy consumption linearly increases with the number of messages in the global buffer independently of how packets are locally distributed, *i.e.,* independently of the number of senders (cf. Sec. 4.3.3).

X-MAC ($B > 2$)

With X-MAC protocol, only two messages can be delivered per each frame. After the transmission of the first data, other devices with buffered messages to send compete among each other (using the extra back-off time) to directly transmit data (without sending any preamble). Nevertheless, the extra back-off time allows the transmission of only one additional data per frame. If buffer size B is larger than 2, at least two frames are needed to empty it. In the following expressions, we assume that no collisions of preambles and messages occur so that the provided expression is rather *optimistic*. Without collision of preambles, it results that frames are always *efficiently filled*, that is, devices always use the minimal number of frames to send B messages.

The computation of $E^X(B)$ uses a modulo operator : if B is even we have to compute the number of *full frames*, *i.e.,* frames during which two messages are sent ; otherwise, if B is odd, the cost of an extra frame for the remaining data must be added. It follows the expression :

$$\begin{aligned} remain(B) &= \quad rem(B, 2) \\ nb_{full\ frames}(B) &= \quad \frac{B - remain(B)}{2} \\ E^X(B) &= \quad nb_{full\ frames}(B) \cdot E^X(2) + remain(B) \cdot E^X(1) \end{aligned} \tag{4.122}$$

Consequently, the evolution of $E^X(B)$ with the increasing values of B is a step function that raises each two messages in the buffer, as depicted in Fig. 4.19.

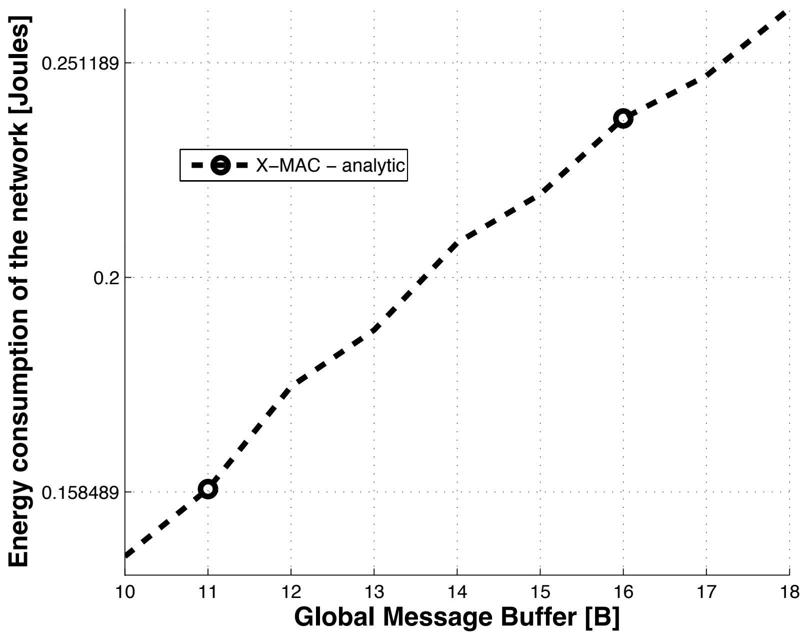


FIGURE 4.19 – Energy analysis for small values of global buffer size. We focus on the model for X-MAC that shows a step trend each two messages in the buffer.

LA-MAC ($B > 2$)

With LA-MAC protocol, several senders can be scheduled per each frame. As for X-MAC, we assume that there are no collisions and that frames are *efficiently filled*, *i.e.,* devices use the minimal number of frames to send B messages.

The limit of data that a frame can contain is fixed by either the duration of a polling period and the duration of a frame, that is the interval between two consecutive wake-ups. Fig. 4.20 shows the organization of an *efficiently filled* frame : after polling period and SCHEDULE transmission, the whole time until next wake-up instant can be used to transmit messages.

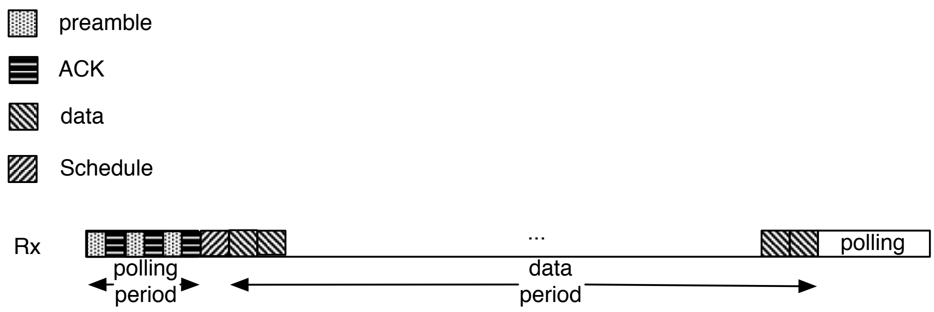


Figure 4.20 – LA-MAC protocol, frame *efficiently filled* with data.

Provided that the polling period has limited duration, the number of preambles that can be cleared during a single polling period is limited as well. For this reason, the way how messages are distributed across nodes influences performance.

Assume that there are 10 messages to send ($B = 10$) and the frame duration is large enough to transmit all messages in a singe frame. If all messages are backlogged in the same buffer, there is only one sender that wakes up the receiver with preambles, receives the SCHEDULE and then transmits all messages to empty its queue. However, if there are 10 senders with one message each, they all try to wake up the receiver ; depending on the collisions that may occur and the limited duration of polling periods only a part of them are cleared by the receiver with an *early* ACK. If this happens, only some senders can transmit during the current frame. All senders that do not receive an *early* ACK go to sleep until the next wake up instant.

In the following expressions, we first assume that each transmitter has a maximum one message in its buffer, then we remove this assumption to provide the final expression.

Such an assumption is rather *pessimistic* for two reasons : first, overhead is high because of the cost of sending a series of preamble is high compared to the benefit of sending a single data and second, if there are B messages in the global buffer, this implies that there will be B contending users that want to send preamble resulting in high traffic congestion.

Analytic expressions that follow assume that energy consumed by all transmitters excepting the first one, is the same. The first sender in fact, is the one who *wakes up* the receiver by sending a series of preambles ; thus, it consumes more than other transmitters that overhear preambles and compete for channel access. With the assumption that each transmitter has only one message to send, we can derive the energy cost of the first transmitter (that is the cost for transmitting the first message) from the expression $E^L(2)$

(cf. Eq. 4.121). In the expression, what is not the consumption of first sender is called *elementary energy consumption* and we assume that this is the cost for each additional transmitter excepting the first one. Let this amount be E^L_{tx2}, we have :

$$E^L_{tx1} = E^L(2) - E^L_{tx2} \tag{4.123}$$

The overall cost of transmission activity depends on the buffer size and elementary energy consumption. We note that E^L_{tx1} and E^L_{tx2} already include the energy cost for the over-hearers.

We now define the maximum number of preambles that can be acknowledged within a single polling process as $nb^{max}_{preambles}$. The maximum number of data messages that can be transmitted within a frame is : nb^{max}_{data}. Because of the assumption that each node can only transmit one message per frame it holds that the maximum number of data that can be delivered within a single frame is limited by the number of preambles that can be sent in the polling period ; such value is $nb^{max}_{data\,per\,frame}$. We have :

$$\begin{aligned} nb^{max}_{preambles} &= \lfloor \frac{t_l}{t^L_p + t^L_a} \rfloor \\ nb^{max}_{data} &= \lfloor \frac{t_f - t_l - t_g}{t_d} \rfloor \\ nb^{max}_{data\,per\,frame} &= min(nb^{max}_{preambles}, nb^{max}_{data}) \end{aligned} \tag{4.124}$$

To compute the number of necessary full frames as well as the number of data in the last and incomplete frame, we use a modulo operator :

$$\begin{aligned} remain(B) &= rem(B, nb^{max}_{data\,per\,frame}) \\ nb_{full\,frames}(B) &= \frac{B - remain(B)}{nb^{max}_{data\,per\,frame}} \end{aligned} \tag{4.125}$$

The overall energy cost is composed of the sum of E^L_{tx1}, a fixed part corresponding to the transmission of the first data, and E^L_{tx2}, additional variable part depending on B and elementary energy consumption. It holds :

$$E^L_{pessimistic}(B) = nb_{full\,frames}(B) \cdot (E^L_{tx1} + (nb^{max}_{data\,per\,frame} - 1) \cdot E^L_{tx2}) + E_{last\,frame}(B) \tag{4.126}$$

where B is used to compute $nb_{full\,frames}$ and $remain$; besides, also last incomplete frame must be considered :

$$E_{last\,frame}(B) = \begin{cases} remain(B) \cdot E^L_{tx1} & \text{if } (remain(B) \leq 1) \\ E^L_{tx1} + (remain(B) - 1) \cdot E^L_{tx2} & \text{otherwise} \end{cases} \tag{4.127}$$

Provided that each transmitter has only one message to send, two situations are possible : either there are few messages in the global buffer so that a portion of polling period of R_x is unused or the number of messages in the global buffer is larger than the maximum number of preambles allowed in a single polling period. We explicit both cases :

- If $(nb^{max}_{data} < nb^{max}_{preambles})$, it means that the receiver spends a part of its polling period without receiving any preamble. For this reason, in this case, we set $nb^{max}_{data\,per\,frame} = nb^{max}_{data}$ and we assume
$$E^L(B) = E^L_{pessimistic}(B) \tag{4.128}$$
- Otherwise, the receiver spends the entire polling period in receiving preambles and sending *early* ACKs. Thus, $nb^{max}_{preambles}$ senders will send one message each. If there are more than $nb^{max}_{preambles}$ messages in the buffer, the senders will need several frames to deliver all of them, thus jeopardizing LA-MAC performance.

We now release the assumption that each sender has only one message to send to derive *optimistic* energy consumption for LA-MAC.

Since $nb_{data}^{max} \geq nb_{preambles}^{max}$, some transmitters will send more than one data message each. We do not need to know how these data messages are distributed across all the different senders.

As previously mentioned, this energy is formed by the part E_{tx1}^{L} for the transmission of the first sender and by several times E_{tx2}^{L}. The total number of messages that are sent in a single frame is nb_{data}^{max}. For each data message, sender and receiver spend t_d seconds respectively in sending and receiving, instead of sleeping. We have :

$$\begin{array}{l} \text{Number of data to send in the last frame :} \\ \quad remain(B) = rem(B, nb_{data}^{max}) \\ \text{Number of complete frames :} \\ \quad nb_{full\ frames}(B) = \frac{B - remain(B)}{nb_{data}^{max}} \end{array} \tag{4.129}$$

$$\begin{aligned} E_{full\ frame}^{L} = & \ E_{tx1}^{L} + (nb_{preambles}^{max} - 1) \cdot E_{tx2}^{L} \\ + & \ (nb_{data}^{max} - nb_{preambles}^{max} + 1) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s) \end{aligned} \tag{4.130}$$

If the buffer size is larger than the maximum number of messages that can be sent in a single frame, is needed an additional frame. The last frame may be not completely filled, either because there are not enough senders to fill the entire polling period, or because there are less than nb_{data}^{max} to send. We have :

$$\begin{array}{l} \text{Number of data to send in the last frame :} \\ \quad remain(B) = rem(B, nb_{data}^{max}) \end{array} \tag{4.131}$$

Thus, energy consumption for the last frame is as follows :

$$\begin{array}{lll} \text{IF} & \multicolumn{2}{l}{(remain(B) \leq nb_{preambles}^{max}) AND (remain(B) \neq 0)} \\ & \text{IF} & (remain(B) = 1) \\ & & E_{last\ frame}^{L}(B) = E^{L}(1) \\ & \text{ELSE} & \\ & & E_{last\ frame}^{L}(B) = E_{tx1}^{L} + (remain(B) - 1) \cdot E_{tx2}^{L} \\ \text{ELSE} & & \\ & \multicolumn{2}{l}{E_{last\ frame}^{L}(B) = E_{full\ frame}^{L} - (nb_{data}^{max} - remain(B)) \cdot t_d \cdot (P_t + P_r - 2 \cdot P_s)} \end{array} \tag{4.132}$$

Finally, we can derive the overall energy consumption :

$$E_{optimistic}^{L}(B) = nb_{full\ frames}(B) \cdot E_{full\ frame}^{L} + E_{last\ frame}^{L}(B) \tag{4.133}$$

Equation 4.133 is *optimistic* for several reasons. First, all frames are *efficiently filled* (cf. Fig. 4.20). The equation assumes that the first $nb_{preambles}^{max}$ that are cleared by the receiver contain a global transmission request so that frames are filled. In the real world however, there is a probability that this not happens : nodes that win the contention and transmit data may have transmission requests of few messages so that frames are not *efficiently filled.* Second, preambles may collide so that even though there are more than $nb_{preambles}^{max}$ senders with backlogged messages, the number of preambles that are cleared in a given polling period is smaller than $nb_{preambles}^{max}$. In this case, some senders must go to sleep and wait for the next wake-up instant. Both *pessimistic* and the *optimistic* expressions are plotted in Fig. 4.21. The curves illustrated in the figure are obtained assuming that $nb_{preambles}^{max}$ is equal to 5 and nb_{data}^{max} to 29. Such values are used in the numerical validation that is presented in the following section (cf. Fig.4.4). As expected, the *pessimistic* curve shows a step trend each 5 messages, because no more than 5 messages can be sent per each frame. In this case, only 5 messages over a maximum of 29 are sent in each frame.

Also the *optimistic* curve shows a step trend, however, in this case the step size is larger, because the optimistic model assumes that frames are always efficiently filled, *i.e.,* there is an increment of consumed energy each 29 messages in the buffer.

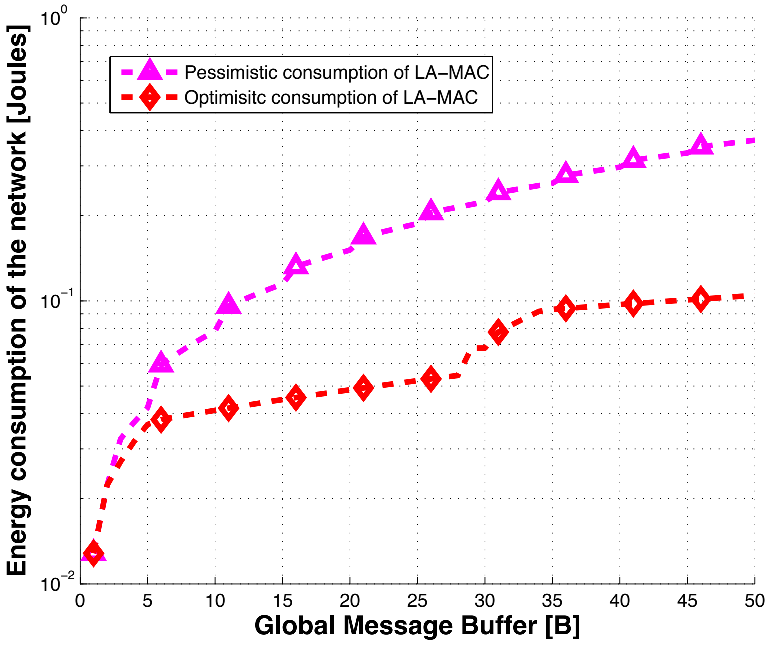


FIGURE 4.21 – Comparison between optimistic and pessimistic energy consumption of LA-MAC vs. the global buffer size.

4.4 Numerical Validation

We have run several simulations using OMNeT++ to validate our energy analysis (cf. Appendix A). Each numerical value is averaged over 1000 independent simulation runs and figures show the corresponding confidence intervals at 95% confidence level. We assume a scenario with $N = 9$ senders and one receiver. The periodical wake-up period is the same for all protocols : $t_f = t_l + t_s = 250\ ms$. Also the polling duration is the same for all protocols : $t_l = 25\ ms$, thus the duty cycle with no messages to send is 10 %. We provide numerical and analytic results for buffer size $B \in [1, 50]$.

In Fig. 4.22, we show the comparison between the proposed energy consumption analysis and numerical simulations for different values of the global buffer size. We assume that at the beginning of each simulation all messages to send are already buffered, so that the first sender starts its channel polling at $t = 0$ and other devices wake up la-

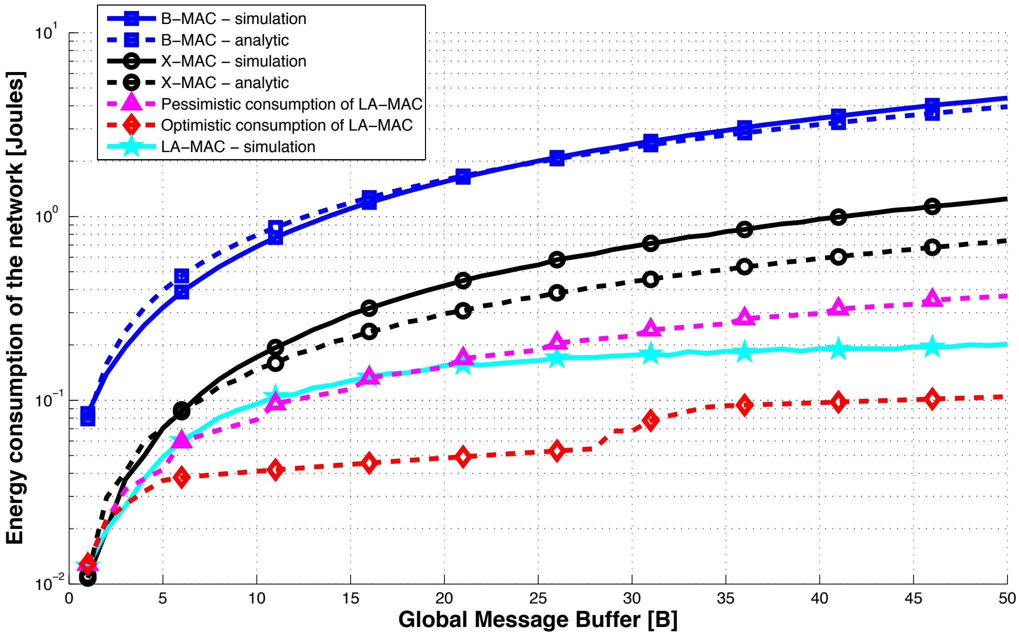


FIGURE 4.22 – Energy analysis and OMNeT++ simulations versus the global buffer size.

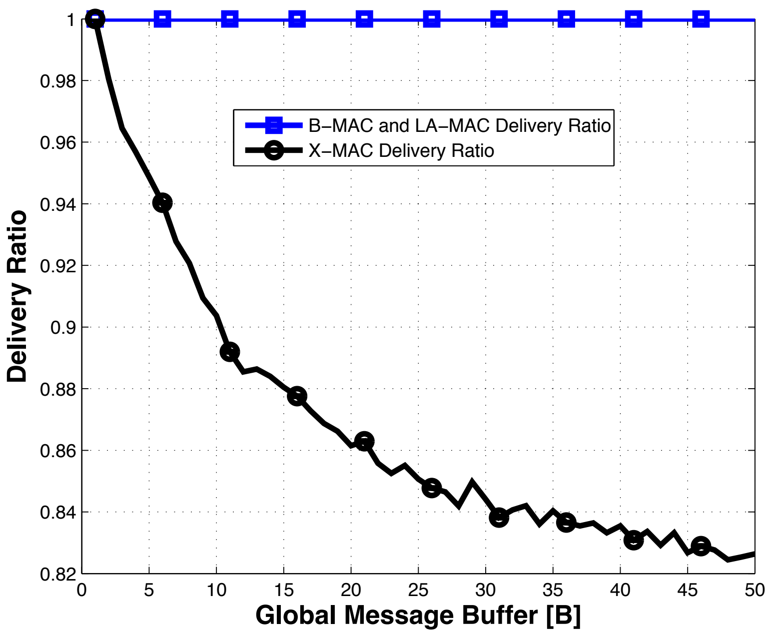


FIGURE 4.23 – Delivery ratio vs. the global message buffer. In X-MAC, most collisions happen when messages are sent after the back-off time.

ter as assumed in the analytic analysis. The simulation stop condition is the delivery of last message in the buffer. Fig. 4.22 highlights the validity of the analytic expressions for energy consumption—we can see that the curves reflect the main trends. The simulation results exceed the analytic data because the simulation reflects the detailed behavior of the protocols, which cannot be captured in simple expressions. As expected, B-MAC is the most energy consuming protocol : as the buffer size increases, the transmission of a long preamble locally saturates the network resulting in high energy consumption and latency (cf. Fig. 4.24). In X-MAC, short preambles mitigate the effect of the increasing local traffic

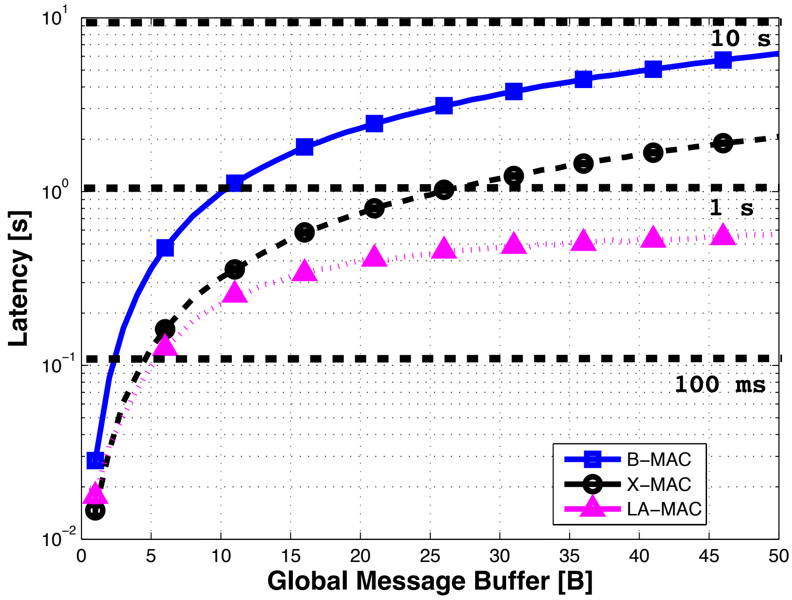


FIGURE 4.24 – Average latency vs. the global message buffer.

load, thus both latency and energy consumption are reduced with respect to B-MAC. Even if X-MAC is more energy efficient than B-MAC, Fig. 4.23 shows that even for small buffer sizes, the delivery ratio for this protocol is lower than 100 % most likely because packets that are sent after the back-off collide at the receiver. Energy consumption of LA-MAC lies in between the *pessimistic* and the *optimistic* curves when global buffer size is higher than 16. When traffic load is light, we observe that energy consumption of LA-MAC slightly exceeds the pessimistic curve. The reason for this is that even though the maximum number of preambles that can be cleared in a polling period is 5 (with current protocol parameters), the probability to clear exactly 5 preambles is low when the number of senders is low. In fact, to clear the maximum of preambles it must happen that one the senders transmits a preamble immediately after the beginning of the polling period of the receiver so that the time between the begin of channel polling and the reception of the first preamble is minimized. When traffic load is light, the number of senders is limited and each node has only few messages to send. Therefore, the probability that there si one of them that sends a preamble immediately after the start of polling process of the receiver is low, resulting in energy consumption similar to the pessimistic case.

In the simulation, all messages in the buffer are distributed among different buffers in a uniform way, so that all cases are possible. Thus, as traffic load increases, the number of senders increases as well so that the probability of having efficiently filled frame becomes higher and energy consumption lies in between the pessimistic and the optimistic curves.

LA-MAC is the most energy saving protocol and it also outperforms other protocols in terms of latency and the delivery ratio. We observe that when the instantaneous buffer size is lower than 2 messages, the cost of the SCHEDULE message is paid in terms of a higher latency with respect to X-MAC (cf. Fig. 4.24) ; however, for larger buffer sizes the cost of the SCHEDULE transmission is compensated by a high number of delivered messages. In Fig. 4.25, we show the percentage of the time during which devices spend in each radio mode versus the global buffer size. Thanks to the efficient message scheduling of LA-MAC, devices sleep most of the time independently of the buffer size and all messages are delivered. Resulting duty cycle (percent of simulation time spent in one of the active

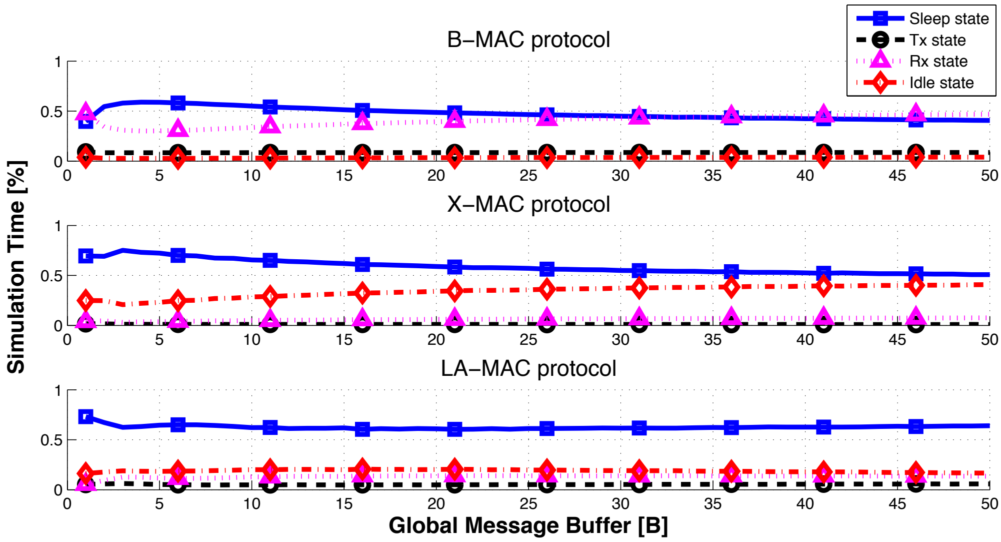


Figure 4.25 – Percentage of the time spent in each radio mode vs. the global message buffer.

modes) is shown in Fig. 4.26. The figure shows that the trend of the duty cycle of LA-MAC differs from the one of B-MAC and X-MAC. The duty cycle trend of B-MAC and X-MAC shows two phases : it first decreases until a value around $B = 3$ and then it increases with traffic load. With LA-MAC, the duty cycle shows a different behavior. It increases with traffic load until a value around $B = 15$, where it reaches its maximum value, and then it decreases.

In B-MAC and X-MAC, the reason for the decreasing phase comes from how the simulation environment is defined. When there is only one or few messages to send, the simulation ends in a short time, that is, as soon as the first sender has finished its transmission. If the simulations is short, the energy consumption of the active couple governs the duty cycle of the entire network. For example, consider the case with B=1. With B-MAC, the sender spends almost all the simulation time in transmission mode (excepting the time that it spends in polling mode before transmitting the preamble) (cf. Tab.4.1a). As consequence, the other nodes *i.e.,* the receiver and the over-hearers spend most of the time in receiving mode because the probability of busy channel when they wake-up is high and they cannot go back to sleep until the end of data transmission. With X-MAC, simulations are shorter with respect to B-MAC resulting in lower duty cycle ; however, the duty cycle shows the same trend.

The simulation duration increases with the value of B. In the second phase of duty cycle, that is when B is larger than 3, we observe that both X-MAC and B-MAC not only result in increasing energy consumption because simulations last more time, but also result in increasing duty cycle. With B-MAC, the duty increases because of the large amount of time that the receiver and over-hearers spend in reception mode. With X-MAC, the duty cycle increases because the number of packets that can be sent in a single frame is limited to two, resulting in high congestion when traffic load becomes heavy.

With LA-MAC, when there is only one message to send, the average simulation duration and duty cycle are in between the duration of X-MAC and B-MAC because of the use of SCHEDULE message. When B increases, the duty cycle increases as well until the maximum of 39.6% that is reached when $B = 15$ (cf. Tab. 4.1c). For values of B lower than 15, the duty cycle of LA-MAC is higher than the one of X-MAC because LA-MAC frames are not efficiently filled, then, the order of the curves is inverted. We observe that even though LA-MAC frames are not efficiently filled, the resulting delivery ratio and latency outperform the values of X-MAC.

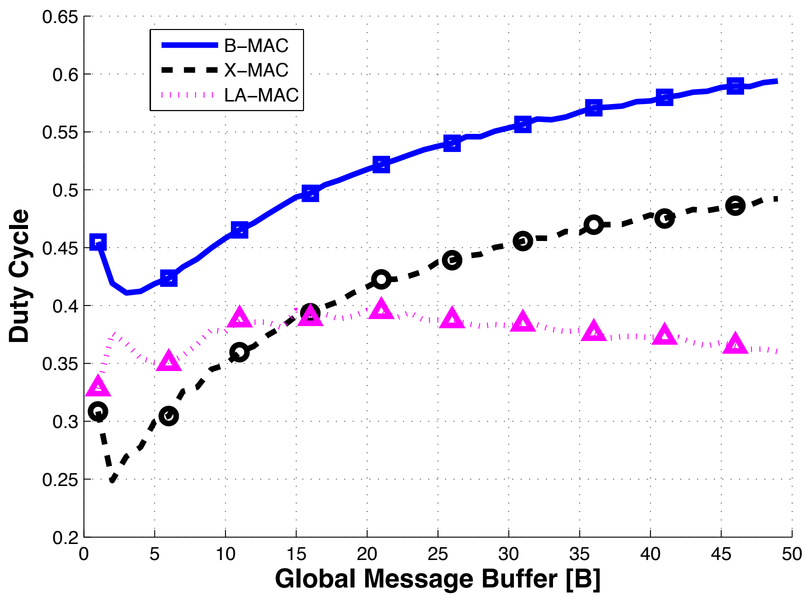


FIGURE 4.26 – Duty cycle vs. the global message buffer.

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.5452	0.0839	0.3422	0.0286	0.4548
3 messages	0.5890	0.0829	0.3013	0.0268	0.4110
5 messages	0.5812	0.0832	0.3080	0.0275	0.4188
15 messages	0.5064	0.0849	0.3761	0.0327	0.4936
30 messages	0.4465	0.0857	0.4313	0.0365	0.5535
50 messages	0.4061	0.0862	0.4686	0.0391	0.5939

(a) B-MAC

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.6915	0.0157	0.0455	0.2472	0.3085
3 messages	0.7304	0.0115	0.0365	0.2216	0.2696
5 messages	0.7003	0.0111	0.0416	0.2470	0.2997
15 messages	0.6090	0.0097	0.0574	0.3239	0.3910
30 messages	0.5477	0.0094	0.0680	0.3749	0.4523
50 messages	0.5078	0.0093	0.0753	0.4075	0.4922

(b) X-MAC

Mode / B size	Sleep mode	Tx mode	Rx mode	Idle mode	Duty Cycle
1 message	0.6717	0.0578	0.0931	0.1774	0.3283
3 messages	0.6328	0.0575	0.1244	0.1853	0.3672
5 messages	0.6491	0.0505	0.1144	0.1860	0.3509
15 messages	0.6043	0.0498	0.1396	0.2063	0.3957
30 messages	0.6175	0.0529	0.1388	0.1908	0.3825
50 messages	0.6399	0.0578	0.1355	0.1668	0.3601

(c) LA-MAC

Table 4.1 – Numerical details of time spent in each radio mode versus the traffic load per different values of B.

4.5 Conclusions

In the present chapter, we analyzed the energy consumption of preamble sampling MAC protocols by means of simple probabilistic modeling. Analytic energy consumption evaluation for large networks is difficult, therefore, we provided an analytic model based on the instantaneous traffic load of a group of contending nodes that want to transmit to the same receiver. The model, easy to understand and apply to preamble sampling protocols, results in accurate energy consumption evaluation with respect to traffic load. With light traffic load, the number of contending nodes is small resulting in low contention and low overall energy consumption. As traffic load increases, the probability of having a higher number of contenders increases as well, resulting in higher contention and consumption. We provided a detailed analytic model for light values of traffic load and a greedy probabilistic expression for heavy values of traffic load.

Simulation results validated the proposed model for a network of 10 nodes ; curves of energy consumption resulting from the simulation fit the curves resulting from the proposed model in the same scenario.

Along with LA-MAC, we applied the analytic model to two classical MAC protocols, B-MAC and X-MAC. Our analysis highlights the energy savings achievable with LA-MAC with respect to B-MAC and X-MAC. It also shows that LA-MAC provides the best performance in the considered case of high density networks under traffic congestion. The proposed analytic model is very flexible and can be used by MAC designers as an approach to understand the energy consumption of PS protocol in different congestion situations.

Chapter 5

Density Aware MAC for Dynamic Wireless Sensor Networks : DA-MAC

The chapter deals with the problem of supporting dynamic network density at the MAC layer. When some devices in the network exhaust their batteries or new nodes are deployed, density distribution varies, resulting in congestion and collision probability variations. Such variations may lead to degraded performance impacting the operation of all protocol layers including MAC and routing. In this chapter, we present a MAC protocol adaptive to density variations. Its principle is to offer a configurable channel sensing phase during which nodes request transmission opportunity in a way that avoids collisions. The receiver can thus schedule transmissions so that nodes may return to sleep and only wake up at their scheduled transmission instants. Allowing burst transmissions improves network capacity and the network can handle load fluctuations. Simulation results shows the performance of DA-MAC compared with two other adaptive access methods : B-MAC [13] with Contention Window and SCP-MAC [17], a protocol that integrates adaptation mechanisms. We consider dynamic wireless sensor networks in which nodes and/or radio links may appear and disappear over the time due to battery exhaustion, propagation conditions, node mobility, or network management operations (e.g. deployment of additional sensors). The goal of DA-MAC (Density Aware MAC) is to find the right trade-off between close synchronization of nodes that enables low duty cycles and avoiding collisions when several nodes tries to access the channel at the same time.

The chapter is organized as follows. Sec. 5.1 focuses on introducing the problem and motivating the proposed solution. Sec. 5.2 presents density heterogeneity and a possible dynamic scenario. Sec. 5.4 presents the problem of adapting channel contention windows with respect to density for collision probability reduction. The proposed DA-MAC design is the subject of Sec. 5.5 followed numerical results (Sec. 5.6) and conclusions (Sec. 5.7.)

5.1 Introduction

We consider *dynamic wireless sensor networks* in which nodes and/or radio links may appear and disappear over the time due to battery exhaustion, propagation conditions, node mobility, or network management operations (*e.g.* deployment of additional sensors). The dynamic behavior may lead to degraded performance in terms of varying connectivity and density impacting thus the operation and performance of all protocol layers. We focus on the MAC layer and propose an access method suitable for dynamic networks in which node density continuously evolves in space and time. Such a MAC must be adaptive and flexible so that the medium access method and frame scheduling algorithms need to take into account traffic load and node density.

The proposed MAC is based on a synchronous duty cycle approach. Its operation is the following :

- Nodes periodically adapt their local protocol parameters to access the channel without collisions.
- Following the synchronous duty cycle approach, each node contends for the transmission of a *burst* of data messages to the next-hop by sending a series of short Request-to-send (RTS) until next-hop node wakes up and it acknowledges RTS reception with a clear-to-send (CTS) message.
- Contending nodes exploit RTS and CTS overhearing and send RTSs using a slotted contention window.
- Neighbor nodes are organized in a structure corresponding to the routing information (a tree, a DAG, a Clustered Tree, or a mesh). In particular, a node knows its parent (a next-hop node) on a given route.
- Potential receiver nodes (parents) periodically wake up each *wake-up interval* and wait for the reception of a series of short preambles for a duration of time equal to the slotted contention window.
- The next-hop allocates the channel to transmitters by sending an ACK frame that defines rendezvous later on for transmitting a burst of data frames.
- A transmitter node goes to sleep, wakes up at the instant of a given rendezvous, at which parent node sends a SCHEDULE message containing transmission organization.
- Transmitters follow the schedule, perform a CCA (Clear Channel Assessment), and send their burst.

The goal of DA-MAC (Density Aware MAC) is to find the right trade-off between close synchronization of nodes that enables low duty cycles and avoiding collisions when several nodes tries to access the channel at the same time. We consider that DA-MAC supports multi-hop networks with convergecast traffic towards a sink. In this case, an intermediate node acting as a relay node can schedule transmission requests of other nodes that need their packets to be forwarded.

DA-MAC adapts to node density by varying the parameters of the channel sensing phase.

5.1.1 Motivations

We consider dynamic wireless sensor networks that are characterized by high variability in terms of density of stations. Density of stations can be expressed as the number of active stations (i.e. stations that need to transmit or relay data packets) per square meter. Rapid density variation may affect network state. Fast density increment results in increasing packet loss due to variation of packet collision. Density reduction may lead to energy waste due to idle listening (energy is wasted listening to the channel without real needs).

Therefore, MAC adaptation in terms of variation of some effective parameters in reaction to density variations can lead to energy savings (major requirement of most WSN), low packet collision and short latency. Moreover, adaptive MAC can also differentiate traffic categories in order to provide differentiated QoS. In the literature for WSNs, density awareness and density control are mostly considered as routing issues. Major solutions switch on and off some nodes to limit and homogenize energy consumption of nodes Chapter 2. As far as MAC protocols are concerned, we are not aware of MAC solutions that adapt operation with respect to density of nodes.

In the following, we define MAC protocol that adapts its behavior with respect to density variations.

5.1.2 Contribution

This chapter is based on a paper presented at the IEEE Personal Indoor Mobile Radio Communications conference [C1] and a patent Pending Patent of Commissariat à l'Energie Atomique [P2].

Density awareness is the key for QoS supporting in density varying sensor networks. To deal with this problem we analyzed the impact of adapting contention window parameters to density variations and we came out with a procedure that keeps constant the contention window size while adapting the behavior of contenders. Moreover, we designed and implemented a novel distributed Medium Access Control (MAC) protocol that uses the adaptation rule to operate in network density varying scenarios. According to our proposed protocol, stations take density as input and are able to adapt some effective parameters in order to match some requirements such as limited energy consumption and a target probability of success. Our contributions are :

- Analytic analysis of contention window variation and definition of adapting procedure,
- Density aware MAC,
- Opening future investigations focused on supporting differentiated traffics coupling the type of traffic and the probability density functions used to choose random contention slots.

5.2 Density Heterogeneity

Network density variations may be the consequence of several events such as random deployment, node mobility, battery exhaustion, node fault, new node deployment or multiple events at the same time. Independently of the specific reason that leads to density variations, the result is that at a given time a WSN may be composed of heterogeneous regions from a density perspective, populated by nodes that do not have the same number of neighbors, *i.e.* that do not experience the same level of interference and congestion. We focus our attention on the effect that such variations imply without considering any specific density variation reason. To study the impact of density variations over a WSN, we designed a network scenario in which density varies over time and space. Density varying scenario is used to evaluate the impact of density and the performance of the proposed MAC protocol.

5.3 System Model

We consider a dynamic wireless sensor network in which the number and the density of nodes may evolve during the network lifetime. As usual, we consider that sensor devices are scattered over a region of interest to monitor the surrounding environment, detect the occurrence of events, and forward messages towards a static *sink* for further data processing. We assume that the network may undergo three phases : *deployment*, *operation*, and *extinction*. During the first phase, nodes are progressively deployed until their place and density is sufficient for the operational phase. When the energy at nodes is exhausted, they stop working, which leads to a progressive extinction.

As a consequence, the resulting network has time-varying characteristics in terms of connectivity and density, which significantly impacts the performance of protocols at all layers. The most important factor from the point of view of the MAC layer is the node density and the current number of neighboring nodes (the nodes that are within the radio range) denoted by N. Actually, the performance of a MAC layer depends on the number of neighbors contending for the channel. We assume that the MAC layer knows the current value of N through exchanging control messages between neighbors [74].

We also assume that the main traffic pattern is multi-point-to-point (MP2P) in which nodes send generated data to the sink in a *multi-hop* way (convergecast). We consider that the network runs RPL [79] to structure the topology as a DODAG (Destination Oriented Directed Acyclic Graph). More precisely, each node has its *rank* (or depth in the DODAG), and it may act as a *leaf*, a *child*, or a *parent* node. Nodes that have the same parent are called *siblings*. Moreover, each sensor node selects the neighbor with the lowest *rank* as its preferred parent (or *next-hop*), the sink being assigned *rank* of 0.

We assume that all nodes except the sink generate *best-effort monitoring* traffic. A node that has some backlogged data to send is said to be *Active* in contrast to *Inactive* nodes.

We mimic density variation with a model in which some nodes exhaust their battery and other nodes are deployed. In the model, the network is composed of a *core sub-network* with a constant number of sensor nodes and a *dynamic sub-network* that goes through phases of deployment and extinction, which corresponds to time-variable node density. Reason for distinguishing two sub-networks is that we are interested in keeping the network always fully connected even though some links may disappear as a consequence of node death. In the model, core sub-network constitutes a fully connected frame above which dynamic sub-network lays on. The model is organized in several phases as plotted in Fig. 5.1 :

Phase 0 : core sub-network deployment :
Between the beginning of network lifetime and time $t = t_0$, is performed the initial node deployment. We assume that our simulations start at $t = t_0$ so that core sub-network is completely deployed (N nodes) and density (average number of neighbor nodes) is constant.

Phase 1 : progressive sub-network deployment :
The network constructs routing tables and in the period from $t = t_0$ to $t = t_1$, nodes belonging to the dynamic sub-network are progressively deployed until reaching the total network size of N' with the increased average density of neighbors. Protocols need to progressively adapt to the new density context.

Phase 2 : constant medium-density phase :
At $t = t_1$, the deployment is finished and the network size is constant until $t = t_2$. Number of nodes and density is constant. Protocols can operate in a constant scenario.

Phase 3 : progressive death of nodes :
At $t = t_2$, nodes of the dynamic sub-network begin to die in a uniform way until $t = t_3$.

Phase 4 : constant low-density phase and sudden deployment :
Between $t = t_3$ and $t = t_4$, the network is again composed of only core nodes. At $t = t_4$, a new deployment phase begins—the density increases instantaneously. Protocols must face a rapidly changing situations and must react to avoid losses due to congestion and collisions.

Phase 5 : constant high-density phase and sudden death :
Until $t = t_5$, the network is composed of N'' nodes, then all dynamic nodes disappear instantaneously causing the density to drop again to the original value and network size is constant (equal to N) again.

Phase 6 : constant low-density phase :
Network population remains constant until the end of simulations at $t = t_6$.

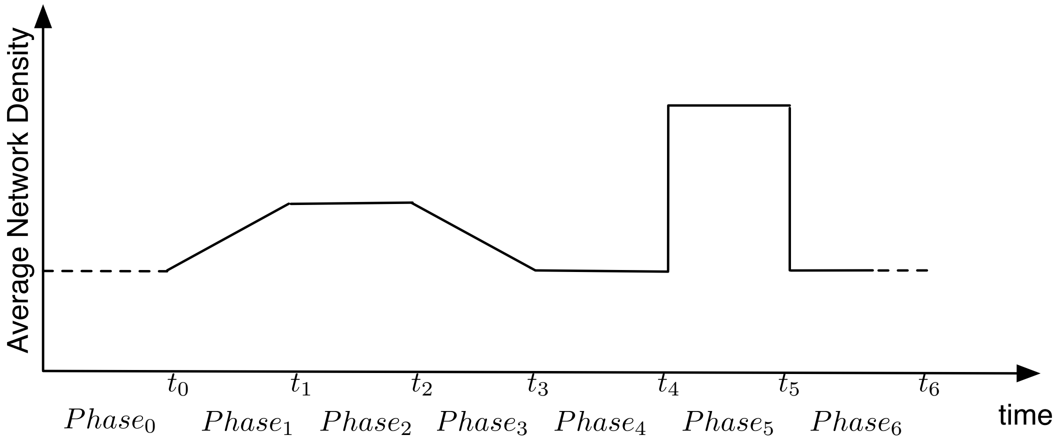


FIGURE 5.1 – Considered dynamic scenario used in numerical simulations.

5.4 Adapting MAC to Density Variations

In particular, we consider synchronous duty cycle MAC protocols that use a constant size contention window to reduce packet collision probability. According to the synchronous duty cycle approach, nodes periodically wake up, contend for channel access, transmit, and then go back to sleep. To contend for channel access, two or more nodes send a transmission request to the receiver ; the receiver acknowledges one of the requests and the winning node can transmit its data. If multiple transmission requests are simultaneously sent to the same receiver, they may collide ; as a result, the receiver is not able to decode any request and cannot acknowledge any of them. Therefore, the period of time during which nodes contend for channel access is a CAP, that is prone to collisions.

At first, we analyze the case of CAP composed of a single contention window CW ; we study the trend of collision probability per different sizes of the window and per different probability distribution functions that are used to select contention slots. Then, we propose to divide the contention period in multiple Contention Rounds (CR_r) with $r \in [1, .., R]$ each one consisting of a small contention window $cw_r < CW = CAP$ and a data transmission slot. The number of rounds within a CAP depends on the size of each small contention windows. We analyze how probability of collision decreases with the number of rounds. Finally, we present the adaptive rule that is the base of the proposed DA-MAC protocol.

5.4.1 Contention Period composed of a Single Contention Window

We consider a network composed of N stations, so that $n \in \aleph[1, .., N]$ represents the station index. The number of active stations N^A that participates in contention is upper bounded by N, but it is lower than N in general. We assume that all stations intend to participate in the contention so that $N^A = N$. All stations that intend to participate to the contention choose one contention slot that initializes a slot counter ; the chosen slot initiates a count-down timer. Stations turn their radio from sleep to poll state until the timer has a value equal to the chosen slot. When the slot counter reaches the selected slot, if the channel is still clear, it transmits data. Otherwise, it sets the count-down timer to NULL, goes back to sleep mode and postpones data transmission to the following contention.

Contention slots are indicated with symbol k, and we assume that $k \in \aleph[0, .., K-1]$. At the beginning of a contention window, all stations simultaneously choose one contention slot so that the entire network slot selection can be represented by the N-sized vector $S = \{S_1, S_2, .., S_N\}$, with $S_i \in \aleph[0, .., K-1]$. We define the event success : *there is only one station that has chosen the minimum slot of S* ; otherwise there is a collision. In other words, if the number of stations that have selected the minimum slot is equal or greater than 2, it means that at least two stations will wake up simultaneously, listen to the channel at the same time and erroneously claim the channel clear for transmission.

At the beginning of a contention window, stations randomly choose their slots according to a given statistical distribution. Station procedure can be resumed as follows :

at the beginning of each contention run

1. Random vector S is generated
2. Collision if : $S_i = S_j$ with $i \neq j \bigwedge S_{k \neq i \neq j} \geq S_i$ for $k = 0, .., K-1$
3. Success for station i if : $S_i = min(S) \bigwedge S_{i \neq j} \geq S_i$ for $i = 1, .., N$

5.4.1.1 Random Uniform Distribution

Given the uniform distribution, the probability that a station chooses one slot among K, with $k \in [0, .., K-1]$ is :

$$\mathbb{P} = \frac{1}{K} \tag{5.1}$$

that is, all slots are equally likely.

When dealing with the probability of success or collision, we can distinguish three different cases depending on which is the event we are interested in. If we take as example the probability of success, we can observe :

1. Probability of success for a reference station.
2. Global probability of success within the contention window.
3. Probability of zero collisions within the contention window.

In the first case, we observe the probability that a user succeeds in the transmission. In practice, if we consider CW, there is no success if another user with respect to the reference one has the minimum slot and the chosen slot is unique. The second case represents the global success probability over the entire group of stations for a single CW. It is a global probability of success meaning that we are interested in the probability of success of anyone among contenders. If compared with the first case, it is always larger because the space of favorable events is larger.

The third case corresponds to what is known as birthday theory or birthday paradox : it is the probability that there is no collision in the entire CW.

In the following, we focus on the evaluation of the first and second cases. The probability of the first case is interesting, because it gives the success information for a given station ; if we consider multiple types of stations, we are interested in knowing what is the probability of success per each member of a given type. The probability of the second case is interesting if we do not distinguish classes or types of stations. In this case, in fact all stations are assumed equivalent and the main goal of the system is to guarantee the transmission to one of them. We leave the third case to future work, because in the present protocol, we are not interested in guaranteeing the absence of collisions. We now consider global probability of success.

Collision Distribution

A collision occurs in a CW if and only if $min(S_i)$ is not unique ; that is, if there are several stations that choose the same slot and that slot is the minimum among the slots of all stations. The probability of a collision differs from the probability of two users choosing the same slot. As already stated, we consider a success when there is only one station that has chosen the minimum slot of S ; therefore, if there is a collision at a slot that has larger value than $min(S)$, the event is still a success. Such an event is not considered as a collision.

A collision occurs at slot k if following two events simultaneously happen :

- Event A : All stations select a slot equal or larger than k
- Event B : There is at least one station that selects slot k or there exists a couple of stations that choose the same slot k

Event A

Event A represents the probability that all stations select a slot equal or larger than k. If we consider N independent stations that adopt uniform random distributions, we have that such probability is the product of N equal elements :

$$\mathbb{P}_A(k) = \left(\frac{K-k}{K}\right)^N \tag{5.2}$$

Event B

Event B represents the probability that at least one station selects slot k or that there exist a couple of stations that choose the same slot k.

Event B is composed of the logical OR of two conditions : B_1 = *at least one station selects slot k* and B_2 = *there exists a couple of stations that chooses the same slot k*.

We can consider $\mathbb{P}_B$ as the complement of $\mathbb{P}_{\overline{B_1}} \cup \mathbb{P}_{\overline{B_2}}$. That is, the complement of the event *zero stations select k* $\cup$ *there is only one station that has chosen k*. Such event can be expressed as :

$$\mathbb{P}_B(k) = 1 - (\mathbb{P}_{\overline{B_1}}(k) + \mathbb{P}_{\overline{B_2}}(k)) \tag{5.3}$$

that leads to :

$$\mathbb{P}_B(k) = \left(1 - \left(\frac{K-k-1}{K-k}\right)^N - N \cdot \frac{(K-k-1)^{N-1}}{(K-k)^N}\right), \tag{5.4}$$

in which N that multiplies the ratio represents all the possible permutations of stations. The complete equation that takes into consideration both $\mathbb{P}_A$ and $\mathbb{P}_B$ is :

$$\mathbb{P}_C(k) = \mathbb{P}_A(k) \cdot \mathbb{P}_B(k) \tag{5.5}$$

so :

$$\mathbb{P}_C(k) = \left(\frac{K-k}{K}\right)^N \cdot \left[1 - \left(\frac{K-k-1}{K-k}\right)^N - N \cdot \frac{(K-k-1)^{N-1}}{(K-k)^N}\right] \tag{5.6}$$

with $k = 0, .., K-1$. Eq. 5.6 can be re-wrtitten as :

$$\mathbb{P}_C(k) = \left(1 - \frac{k}{K}\right)^N \cdot \left\{1 - \left[1 - \left(\frac{1}{K-k}\right)^N\right] - N \cdot \left[\frac{1}{K-k} \cdot \left(1 - \frac{1}{K-k}\right)^{N-1}\right]\right\} \tag{5.7}$$

The collision distribution for a network population of N=10 and K=30 is illustrated in Fig. 5.2. We can see that the collision distribution is a decreasing function of k. The reason for that is that if we consider large values of k, the probability of having a large number of stations that choose larger k decreases until it reaches zero at k=K. The figure also shows a Monte-Carlo simulation ($100 \cdot 10^3$ iterations) for N=10 and K=30 that fits the analytic equation.

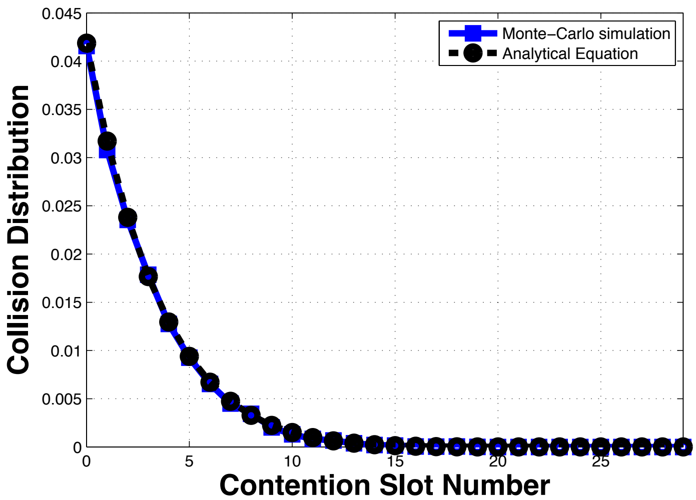


Figure 5.2 – Analytic equation and Monte-Carlo simulation for collision distribution vs. contention slot number. Contention slots are chosen following the uniform distribution. *N=10, K=30.*

Fig. 5.3 illustrates the probability of collision as a function of contention window size K for three different network sizes N=*10,30*, and 50. As expected, the probability of collision decreases with CW size and increases if network size increases.

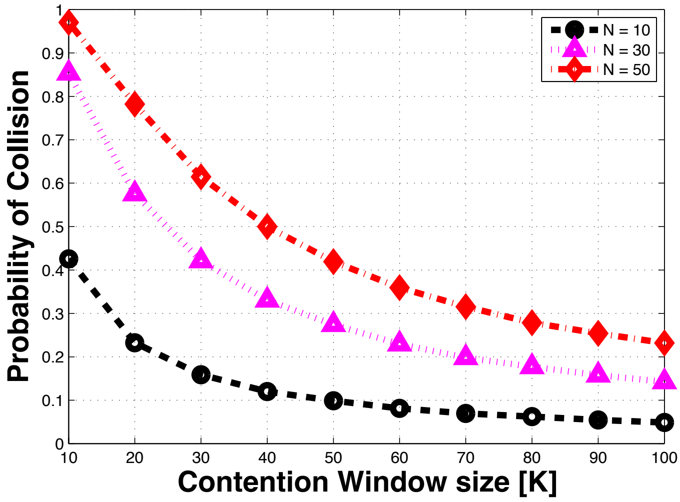


FIGURE 5.3 – Probability of collision vs. contention window size K when access time slots are chosen following the uniform distribution. Contention window size varies in the range [10,..,100].

Success Distribution
The event of success occurs when *min(S)* is unique. In terms of probability, we can define the probability of success as *the probability that there is only one station that chooses* k *and that all remaining N-1 choose a different slot*, that is :

$$\mathbb{P}_{s,u}(k) = N \cdot \frac{1}{K} \cdot \left(1 - \frac{k}{K}\right)^{N-1} \tag{5.8}$$

By running a Monte-Carlo simulation with $100 \cdot 10^3$ iterations, we show that Equation 5.8 represents the distribution of success for each slot k (see Fig. 5.4).
Fig. 5.4, shows the distribution of success over k as a result of a simulation for N=10 K=30.

The distribution of success per slot is a decreasing function (as well as the distribution of collisions), in fact it is reasonable to assume that the higher the selected slot, the lower the probability that all other stations choose a larger slot. Fig. 5.5 illustrates the probability of success as a function of contention window size K.

5.4.1.2 Generic Random Distribution

If the random distribution is not uniform, the analytic expression for the collision and success distribution become more complex. A generalized expression for the probability of success of a given station at slot k is :

$$\mathbb{P}_{s,g}(k) = \sum_{n=1}^{N} \mathbb{P}(S_n = k) \cdot \prod_{\binom{i=1}{i \neq n}}^{N} [\mathbb{P}(S_i = k+1) + \mathbb{P}(S_i = k+2) + .. + \mathbb{P}(S_i = K)] \tag{5.9}$$

that is the sum of the probabilities of success of all stations. If we are interested in the probability of success for a given station $S_{\overline{n}}$, we have :

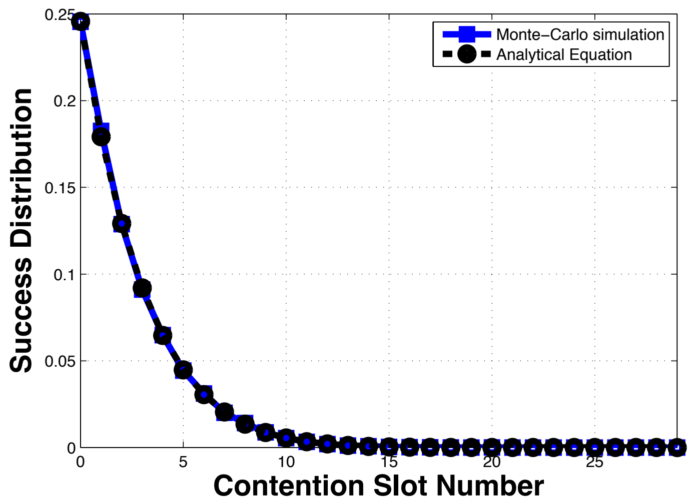


Figure 5.4 – Analytic equation and Monte-Carlo simulation for success distribution vs. contention slot number. Contention slots are chosen following the uniform distribution. *N=10, K=30.*

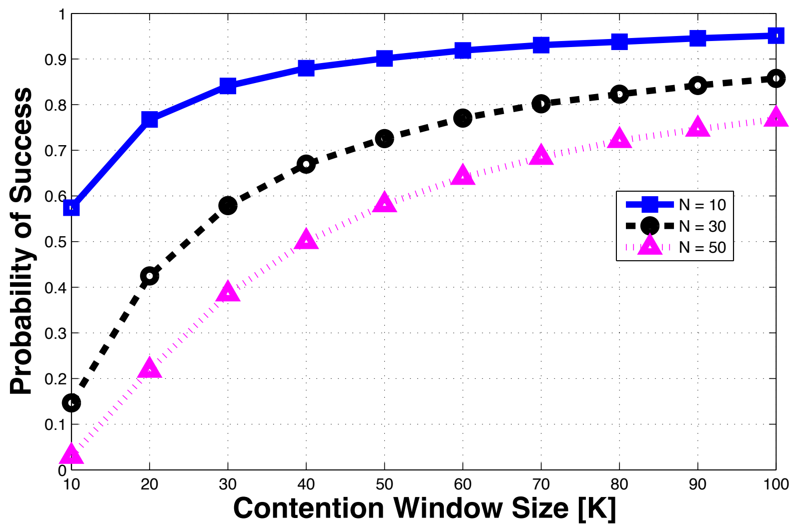


Figure 5.5 – Probability of success vs. contention window size K when access time slots are chosen following the uniform distribution. Contention window size varies in the range [10,..,100].

$$\mathbb{P}_{s,S_{\overline{n}}}(k) = \mathbb{P}(S_{\overline{n}} = k) \cdot \prod_{\binom{i=1}{i \neq \overline{n}}}^{N} \left[\mathbb{P}(S_i = k+1) + \mathbb{P}(S_i = k+2) + .. + \mathbb{P}(S_i = K)\right] \tag{5.10}$$

5.4.1.3 Negative Exponential Distribution

The shape of the negative exponential distribution depends on the value of a single parameter μ. The probability distribution function is as follows :

$$pdf_{exp}(x;\mu) = \begin{cases} \mu \cdot e^{-\mu \cdot x}, & \text{if } x \geq 0\,; \\ 0, & \text{x<0.} \end{cases} \tag{5.11}$$

and cumulative distribution function :

$$CDF_{exp}(x;\mu) = \begin{cases} 1 - e^{-\mu \cdot x}, & \text{if } x \geq 0\,; \\ 0, & \text{x<0.} \end{cases} \tag{5.12}$$

To select random slots according to the negative exponential distribution, we approximate its cumulative distribution function to define a look up table that associates a random number in the range (0,1) to an integer $\in [0, K-1]$, with K being the size of the contention window. As an example, in Fig. 5.6 we present the look up table corresponding to the case of K=10 and for different values of the shape factor μ.

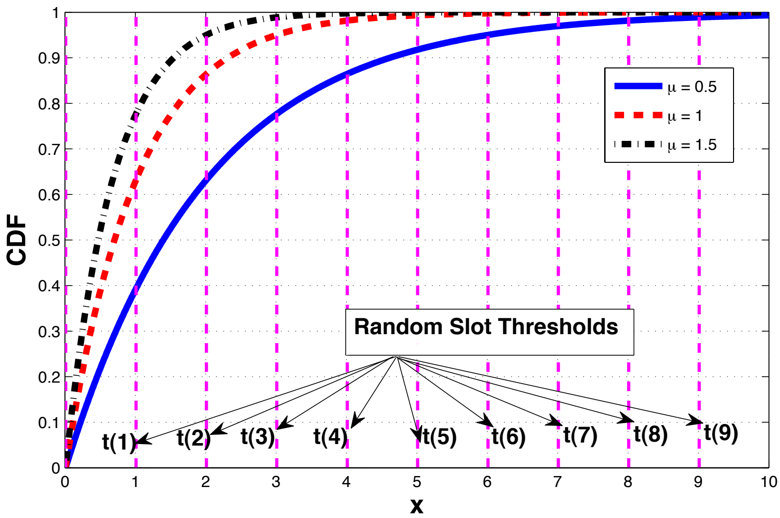


FIGURE 5.6 – Look-up table for the random slot allocation following the negative exponential distribution. Example with K=10 (contention slots lie in [0,9]) for three shape factors.

The rule to choose random slots in the range [0,K-1] following the exponential distribution of parameter μ is as follows :

Step 0 Requirements : per each user i, μ', K
Step 1 F(x) = $CDF\{x, \mu'\}$,
Step 2 r=*uniform rand(0,1)* ; k = 0
Step 3 WHILE(r > $F(t(k+1))$) ; k = k +1 ; END WHILE

Fig. 5.7 illustrates random slot allocation following the negative exponential distribution when CW size is K=30 for three different values of μ. The same figure we also shows the distribution of slots that follows the uniform distribution.

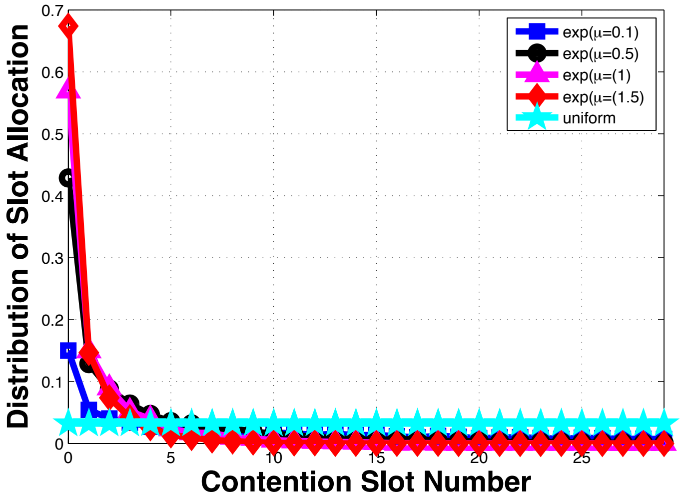


Figure 5.7 – Random slot allocation following negative exponential distribution. The size of the contention window is K=30 (contention slots lie in [0,29]).

.

Increasing μ has the effect of increasing the probability of selecting small contention slots.

Collision Distribution

Fig. 5.8 illustrates the distribution of collisions for the scenario N=10, K=30 for three different values of μ. From a comparison with the uniform distribution, we can see that we can shift the probability of collision just by shaping slot distribution rule. If several nodes use the exponential random distribution with the same value of μ, they have high probability to choose the same slot. Moreover, such a probability increases with the value of μ. As a consequence, the success distribution also changes as described below.

As far as the probability of collision is concerned, we show different curves varying CW size [K] in Fig. 5.9. As expected, the probability of collision is higher for higher values of μ, because when increasing μ, there is a higher concentration of stations around small slots.

Success Distribution

The distribution of successes has a similar decreasing behavior as the distribution of collisions (cf. Fig. 5.10).

We show the probability of success of the exponential distribution in Fig. 5.11. From the figure we observe that letting all stations to select a random slot with the same distribution is not a good choice because even though CW size is large, the resulting success probability is lower than the probability of success of the uniform distribution.

5.4.1.4 Beta Distribution

To provision differentiated QoS to different stations, we now investigate the Beta distribution. It is a family of continuous probability distributions defined on the interval (0, 1) parameterized by two positive shape parameters, denoted by α and β.

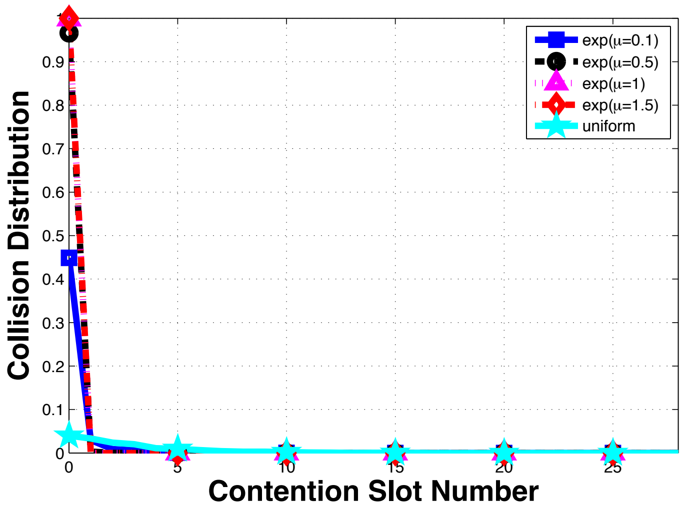


FIGURE 5.8 – Distribution of collisions when random slots are allocated according to the negative exponential distributions. N=10, K=30 (contention slots lie in [0,29]).

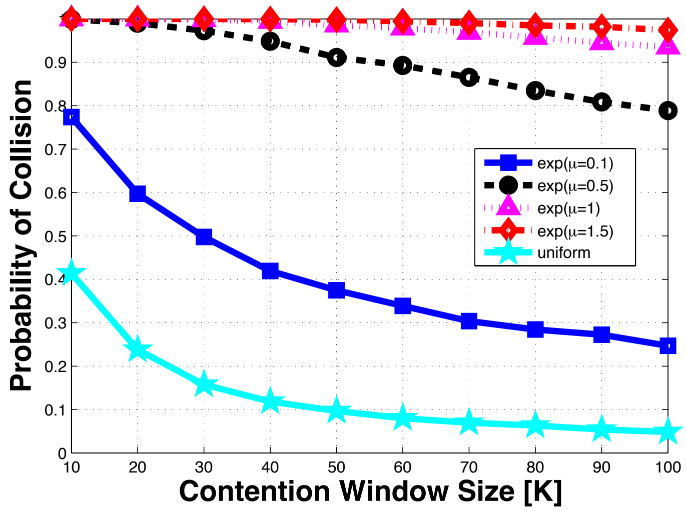


FIGURE 5.9 – Probability of collision of the negative exponential distribution as a function of contention window size K, N=10. Comparisons for different values of μ.

The distribution follows the equation :

$$pdf_{Beta}(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha;\beta)} \tag{5.13}$$

in which $B(\alpha;\beta)$ is a normalization term. The Cumulative Distribution Function is expressed by the regularized incomplete Beta function $I_x(\alpha,\beta)$. Its expression is as follows :

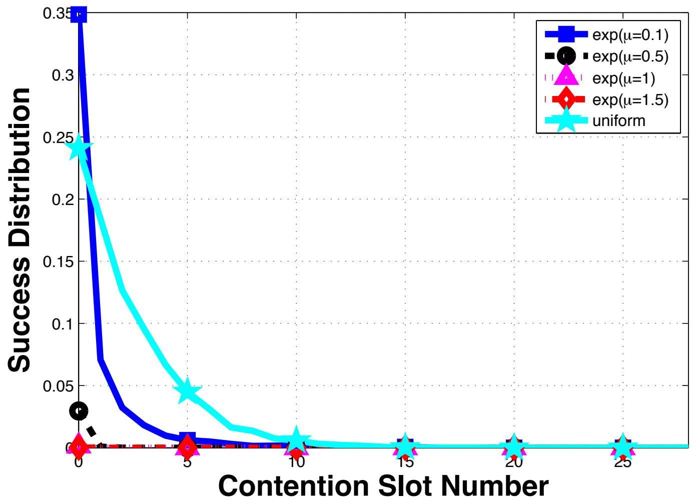


FIGURE 5.10 – Distribution of success when slots are allocated following negative exponential distributions. N=10, K=30 (contention slots lie in [0,29]).

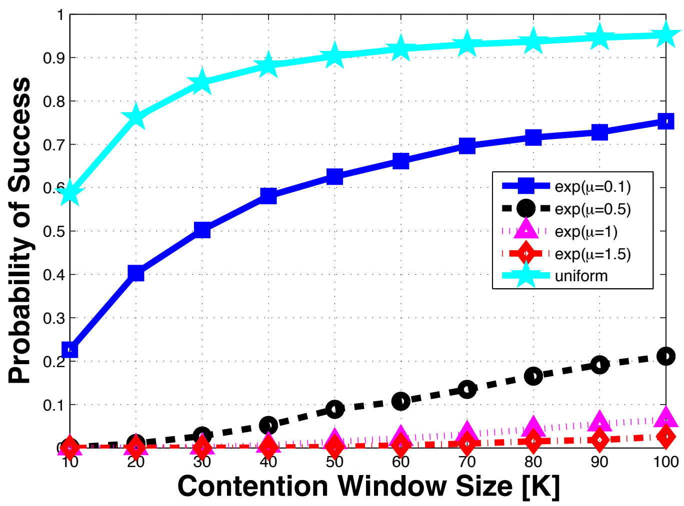


FIGURE 5.11 – Probability of success for the negative exponential distribution as a function of contention window size K. Comparisons for different values of μ.

$$
\begin{aligned}
CDF_{Beta}(x;\alpha,\beta) &= \\
I_x(\alpha,\beta) &= \sum_{j=\alpha}^{\alpha+\beta-1} \frac{(\alpha+\beta-1)!}{j!(\alpha+\beta-1-j)!} \cdot x^j \cdot (1-x)^{\alpha+\beta-1-j}.
\end{aligned} \tag{5.14}
$$

The shape of the Beta distribution varies with the shape parameters. Following the same approach as in the case of the negative exponential distribution, we approximate the CDF of the Beta distribution to define a look-up-table for random slot allocation.

We present the random slot allocation following the Beta distribution for different shape factors in Fig. 5.12. The distributions of success and collision will be investigated in the following section.

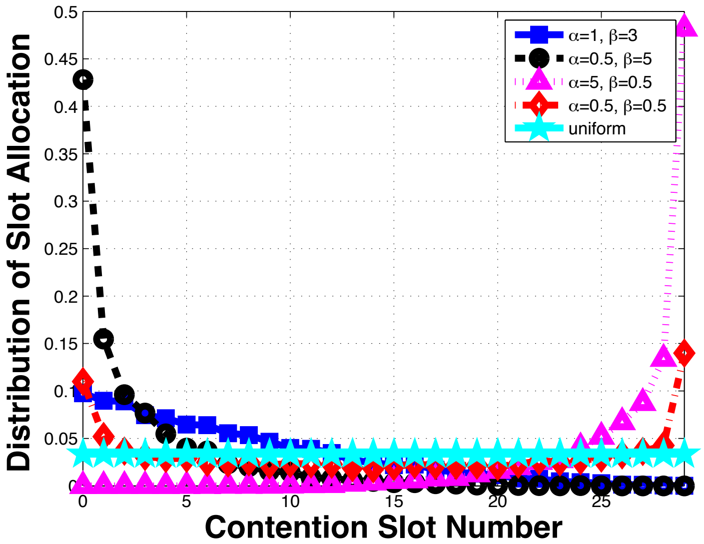


FIGURE 5.12 – Slot allocation distribution with a contention window of size K=30 for several shape factors.

QoS Provisioning : Coupling Different Traffic Priorities and Random Distributions

We consider a network composed of N stations. If we assume that one of them has urgent data to transmit, we want to differentiate traffic and give it higher priority.

Fig. 5.13 and 5.14 show the collision and success distributions for the following three differentiation scenarios :

1. No traffic differentiation : all stations adopt the uniform distribution.
2. Simple Traffic differentiation : urgent stations choose a slot according to the Beta distribution $(\alpha = 0.5, \beta = 5)$; non urgent stations adopt the uniform distribution.
3. Extreme Traffic differentiation : urgent stations choose a slot according to the Beta distribution $(\alpha = 0.5, \beta = 5)$; non urgent stations adopt the Beta distribution with different shape parameters $(\alpha = 5, \beta = 0.5)$

We can see that by discriminating different types of traffic, we are able to give priority to desired stations. Fig. 5.15 and 5.16 show the shape of the probability of collision and success for the same scenarios as contention window size K increases.

We observe that in scenario 2, where there is only one station that adopts the Beta distribution and others adopt the uniform distribution, the probability of collision is worse than in scenario 1 (all stations adopt the uniform distribution) ; another difference between scenario 1 and 2 is the way how collision are distributed over K contention slots (as illustrated in Fig. 5.13).

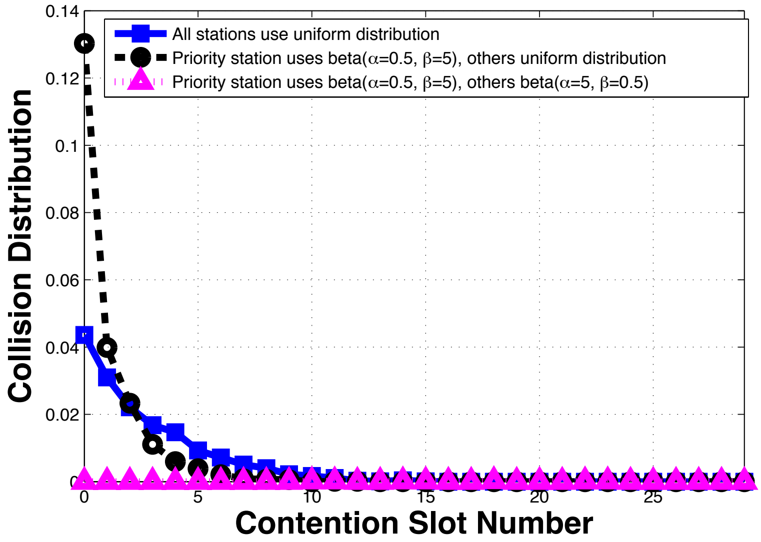


Figure 5.13 – Distribution of collision when random slots are allocated following the Beta distribution. Comparisons of several differentiation scenarios for N=10, K=30.

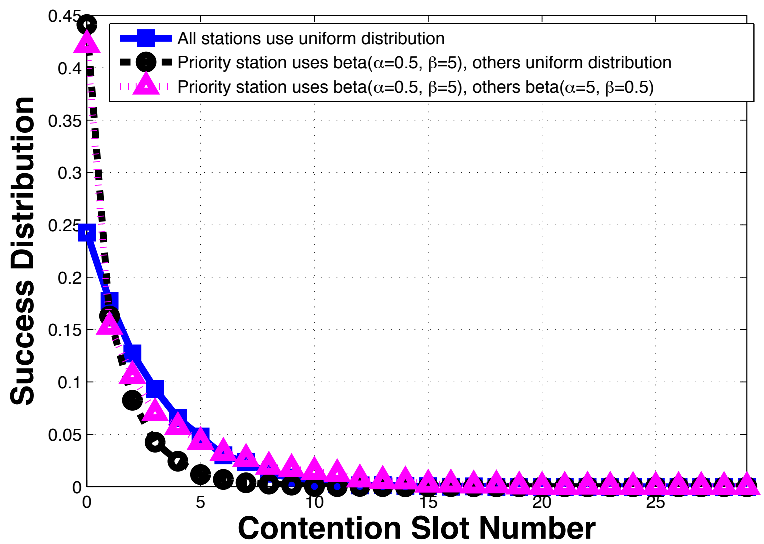


Figure 5.14 – Distribution of success when random slots are allocated following the Beta distribution. Comparisons of several differentiation scenarios for N=10, K=30.

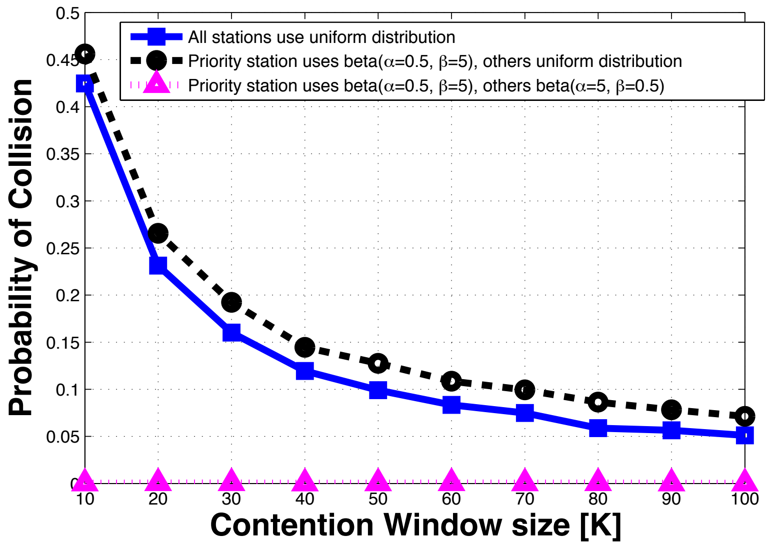


FIGURE 5.15 – Probability of collision when random slots are allocated following the Beta distribution. Comparisons of several differentiation scenarios for N=10, K=30.

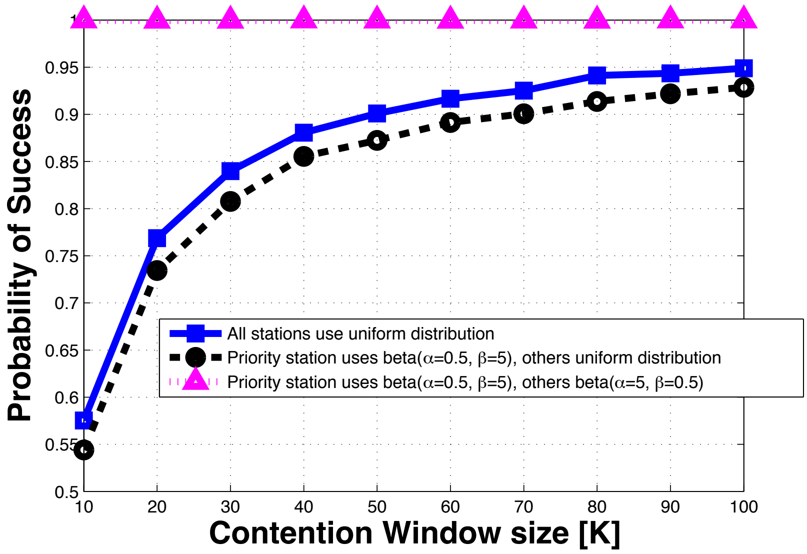


FIGURE 5.16 – Probability of success when random slots are allocated following the Beta distribution. Comparisons of several differentiation scenarios for N=10, K=30.

5.4.2 Contention Period composed of Multiple Contention Rounds

We now assume that the constant size CAP is structured as a series of Contention Rounds. Each Contention Round (CR_r) with $r \in [1, .., R]$ consists of a small contention window $cw_r < CW$ and a data transmission slot δ ; thus, the number of CR within a CAP depends on the size of each small contention windows (cf. Fig. 5.17).

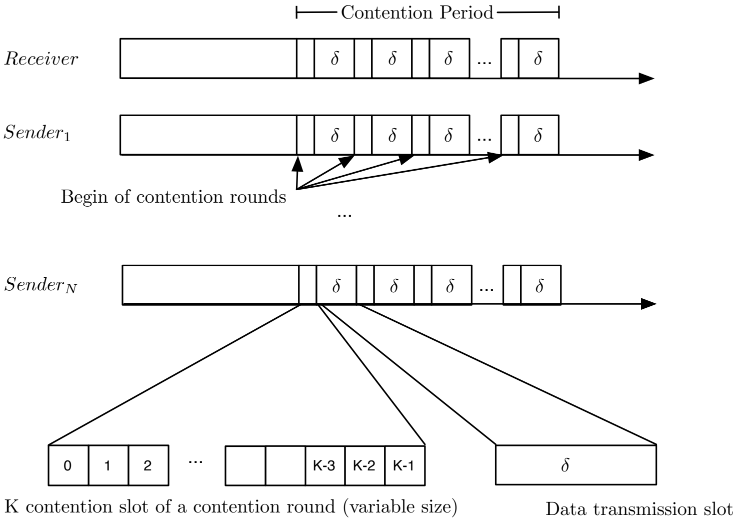


Figure 5.17 – Contention Access Period consisting of a series of consecutive Contention Rounds.

Each short CR is composed of a small contention window cw of size K (K slots of duration γ) and is followed by a period of time (indicated as δ) for data transmission. During δ the winning stations will transmit their data packets. If the winning station is unique, data transmission succeeds, otherwise a collision may occur.

If we consider a contention round as a single contention window, the probability of success and collision depend on the number of contenders, the size of the contention window and the probability distribution function that is used to select each contention slot. Nevertheless, since the contention period is composed of a series of CRs, the probability of success of a station within a CAP changes with respect to the case of a single contention window. In fact, the probability that a node is successful in R rounds is the sum of R elements of a geometric distribution with trial success probability P_s, where P_s is the probability of success over a single round and that depends on the particular probability distribution function. We have :

$$\mathbb{P}_s^R = \sum_{r=1}^{R} \mathbb{P}_s \cdot (1 - \mathbb{P}_s)^{r-1} \tag{5.15}$$

5.4.3 Adaptive Contention Window in function of Density of Contenders

We propose a density aware MAC protocol that adopts multiple consecutive Contention Rounds with a fixed size CAP. We assume that contending stations are aware of the number of neighbor nodes that may interfere with transmission ; such value can be periodically estimated by each station as well as received by a coordinator node who periodically estimate it. Stations are assumed synchronized, that is, their wake-up schedules are the closely synchronized. Before each contention, a sender adapts its local parameters to the last known value of neighbor density. The adaptation rule aims at defining two parameters :

- The value of K for the current CAP,
- P_p, probability of participation.

The value of the parameter K, sets the number of slots that compose each contention round of a given CAP. All contention rounds in fact, have the same size, that is, during a given CAP all stations contend the channel using CWs that have the same size. Provided that CAP has a fixed size and it is constant, the higher the size of CW, the smaller the number of contention rounds (see Fig. 5.17). The number of contention rounds is indicated by R. $\mathbb{P}_p \in \Re \bigwedge \mathbb{P}_p \in [0,1]$ is the probability of participation ; that is the probability according to which a station will participate in the contention for a given CAP. If a node decides to not contend, it goes back to sleep until the next contention period. The higher value of $\mathbb{P}_p$, the higher the probability of participation. When the network is not much populated, the optimization algorithm provides high $\mathbb{P}_p$ and small CW size. Adaptation is subject to $\mathbb{P}_t$ that is the target probability of success set up by application requirements. Adaptation steps are provided for the case of the uniform probability distribution function :

Step 0 requirements $\mathbb{P}_t$, N and size of CAP.
Step 1 init $\mathbb{P}_p$=1, K=$default_k$, $R = default_R$
Step 2
Evaluate $\mathbb{P}_s = \sum_{m=1}^{K} \mathbb{P}_p \cdot N \cdot \frac{m}{K} \cdot (1 - \frac{m}{K})^{(N-1)\cdot \mathbb{P}_p}$
Evaluate $\mathbb{P}_s^R = \sum_{t=1}^{R} \mathbb{P}_s \cdot (1 - \mathbb{P}_s)^{(r-1)}$
WHILE : $\mathbb{P}_s < \mathbb{P}_{target}$
FOR : $R = default_R$ to 1
FOR : $\mathbb{P}_p$=1 to 0.1
FOR : K=$default_k$ to max_k
Evaluate $\mathbb{P}_s$
IF $(\mathbb{P}_s^R \geq \mathbb{P}_t)$:found $K, \mathbb{P}_p, R$
ELSE : go on
ENDFOR
ENDFOR
ENDFOR
ENDWHILE
Step 3 RETURN (K, P_p, R)

where $default_k$ and $default_R$ depend on application layer constraints.

Thank to the density adaptation rule, contending stations can successfully transmit with a probability higher than a target value.

The adaptation rule presented above, is a independent of any wireless access protocol. In the following section, we propose a novel density aware MAC protocol for WSNs that makes the use of the adaptation rule. The proposed protocol, results in efficient transmission in WSNs with heterogeneous density distribution.

5.5 Proposed DA-MAC method

We now specify the principles of the proposed density aware MAC.

5.5.1 DA-MAC operation

DA-MAC aims at operating at very low duty cycles for energy efficiency, which requires close synchronization of wake up instants. At the same time, we need to avoid collisions when several contending nodes wake up and try to transmit at the same time. Moreover, in a multi-hop network with convergecast traffic, parent nodes may need to forward packets of their children so when a receiver is already waken up, they would benefit from the possibility of transmitting several frames.

The principle of DA-MAC is to start with a channel polling phase in which *Active* sibling nodes request transmission opportunities and the parent defines a schedule for transmissions (at which instant a given node needs to start its transmission and for how long). Obtaining low duty cycles requires close synchronization between nodes, so when they wake up and request transmissions at the same time, this may result in excessive collisions. To avoid this situation, we define the sensing phase as a sequence of several *contention rounds*, each round with collision avoidance through a randomized back-off. DA-MAC adapts to node density by varying the parameter of the channel polling phase, mainly the number of contention rounds and the randomized contention window in function of N, the number of contending neighbors.

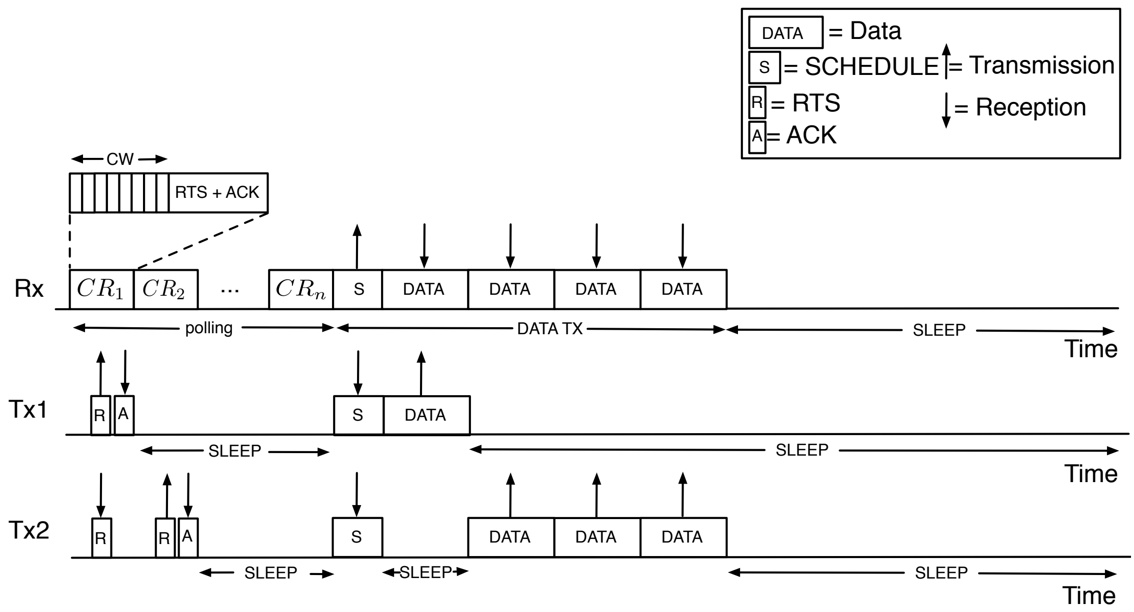


FIGURE 5.18 – Principles of DA-MAC operation.

Fig. 5.18 illustrates the principles of DA-MAC operation—it is composed of three periodical phases : channel sensing, a data transmission period of variable duration, and a long sleep period also of variable duration. Block scheme of protocol operation are illustrated in Fig. 5.19.

Step 1 *: Wake up and poll the channel.*

Channel sensing of constant duration t_l is composed of a sequence of R Contention Rounds (CR), each round $r \in [1, .., R]$ containing a Contention Window (CW) of K slots and an interval of constant duration for an exchange of Request To Send (RTS) and Acknowledgment (ACK) messages.

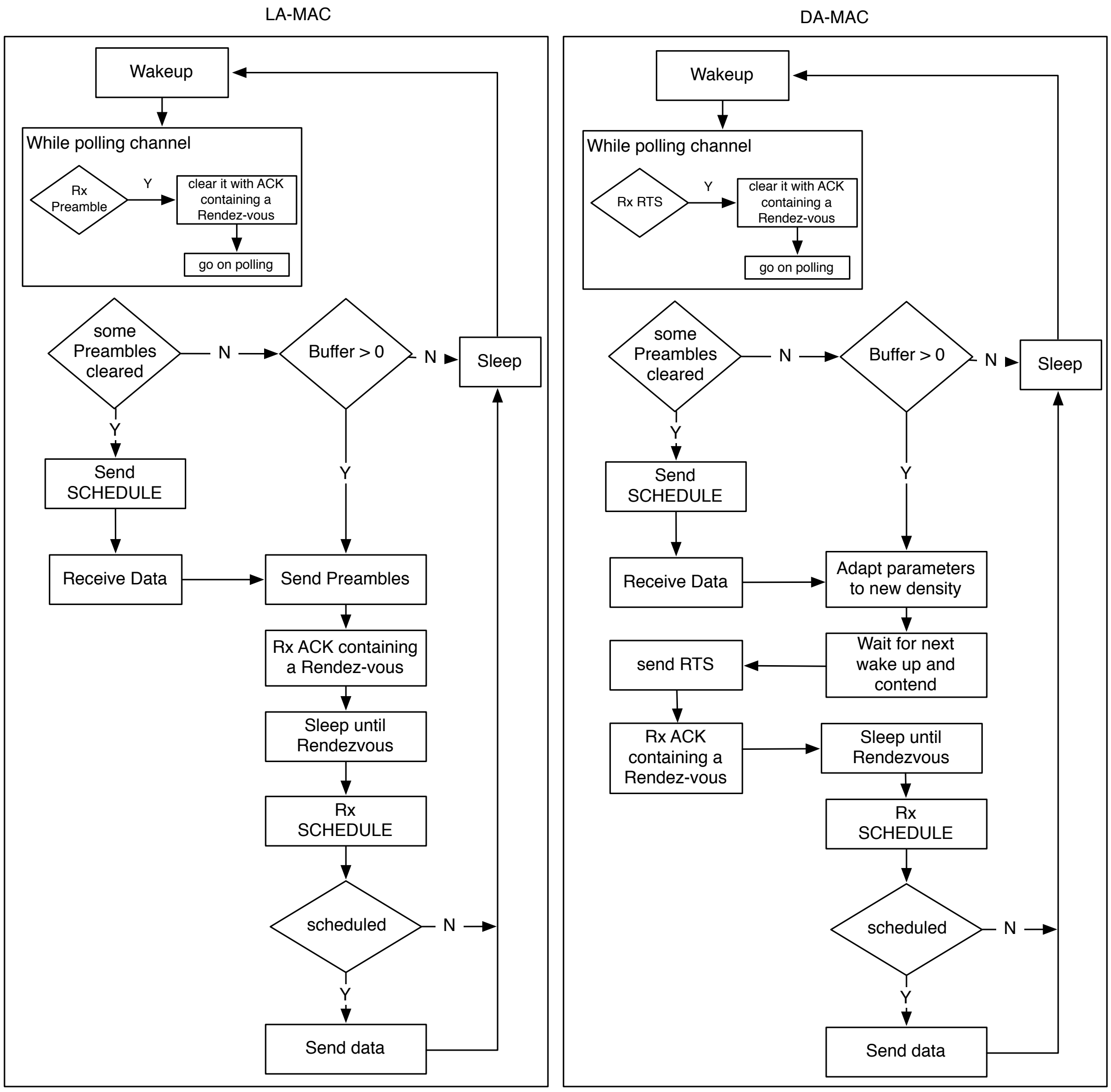


FIGURE 5.19 – Block diagram for LA-MAC and DA-MAC protocols. Diagrams refer to the case of unicast data frame transmission.

An *Active* node contends for channel access with probability $\mathbb{P}_p$, a parameter adjusted in function of node density. If it contends, it will try to send an RTS during CR_1. If it does not receive an ACK, it tries again in the next round until an ACK is received or channel polling ends. In a round, a node chooses a uniformly distributed slot $k \in \aleph[0, .., K-1]$ and senses the channel until the chosen slot. If channel is clear during this interval, the node transmits its RTS message with the following information : the request for a burst of data (one or more packets) to send and the current estimated number of neighbors N. Nodes that overhear the RTS frame consider it as a blocking signal so they defer until the start of the following round. If the RTS message comes from a parent node requesting to send data to its own parent, the current node interprets it as a do-not-disturb signal and goes to sleep until the next channel sensing phase starts.

When a node correctly receives an RTS (no collision), it "clears" it by sending an ACK frame with the instant of a *rendezvous* when it broadcasts a SCHEDULE frame with the result of its scheduling decision. The ACK also forces the transmitter to stop contending so that another node can transmit its RTS with success during the following round. After receiving an ACK, the transmitter goes to sleep until the instant of the rendezvous. After sending an RTS, nodes set up a timeout equal to the *CR* period. If they do not receive an ACK, they try to participate again.

Step 2 *: Broadcast of the SCHEDULE frame.*

After the channel polling period, the receiver has all the requests of transmitters so it can define the schedule for data transmissions, which is based on the First Come First Served (FCFS) rule (an extension to handle different traffic priorities is straightforward). If the transmission requests exceed the maximum available transmission time for the current wake up period, last requests are ignored. The receiver sends a broadcast SCHEDULE frame that includes : the list of instants at which transmitters can transmit their bursts, the MAC address of each transmitter, the instant of the transmission end, the number of neighbors estimated by children (2-hop node density).

Step 3 *: Burst transmission.*

After receiving the SCHEDULE frame, children nodes transmit their bursts at the defined instants. While waiting for its turn, a child can go to sleep and wake up at the instant of its transmission. If a node needs to broadcast a frame or a burst, it can mark its burst as a broadcast in the requesting RTS. In this case, the receiver schedules all children (and itself) to be awaken during the transmission so that all nodes will receive the broadcast.

Step 4 *: Next wake-up period.*

After transmissions, nodes go to sleep and wake up at the instant of the next channel sensing period.

5.5.2 Adapting Parameters to Node Density

As each node transmits the number of neighbors in its transmission range to its parent, nodes are aware of the node density in the vicinity. A node chooses the maximum between its estimated number of neighbors and the number reported by its children to be sure that the access scheme has sufficient configuration to avoid collisions during channel sensing phase.

Let us consider that a generic node has $N-1$ contenders participating in channel sensing with probability P_p ; if each contender has the same probability of choosing one slot among K, the probability of a successful RTS/ACK exchange on a single round and the probability of success after R rounds are $\mathbb{P}_s$ and $\mathbb{P}_s^R$, respectively (cf. Sec. 5.4.3).

5.5.3 Synchronization

We assume that nodes are able to synchronize on a given schedule so they can wake up at a given instant and cope with clock drift by transmitting dedicated synchronization messages or by piggybacking synchronous information onto data messages like in SCP-MAC [17]. In our case, we assume the network topology structured in the form of a DODAG that can be used for propagating the synchronization from the sink to leaves.

DA-MAC requires that all contending nodes know the instant of the start of channel sensing. This information is embedded in the SCHEDULE message, so when a new node joins the network, it must listen to the channel until it receives the first SCHEDULE. For this reason even if any RTS is collected each node that is already synchronized to the network must periodically send an empty SCHEDULE message to advertise the beginning of its next channel sensing interval ; such a period depends on applications requirements.

5.6 Simulation Results

5.6.1 Simulation Environment

Numerical simulations to analyze the performance of the proposed DA-MAC protocol are performed using the OMNeT++ simulator (cf. Appendix A). We compare the performance of three MAC protocols : B-MAC with a Contention Window [13, 17], SCP-MAC [17], and proposed DA-MAC in two different scenarios. We have chosen B-MAC and SCP-MAC for the comparisons because SCP-MAC adjusts protocol parameters to take into account changing network conditions and B-MAC is a reference protocol for WSNs that outperforms synchronous duty cycle protocols [77]. Operations of the three protocols are compared in Fig. 5.20.

We simulate sensor networks of different sizes $\mathrm{N} \in [30, 100]$ nodes during 10000 s and the simulation results are averaged over 10 runs (figures present the results with corresponding confidence intervals). At each run, nodes are randomly placed at different locations. Each time a new node is deployed or dies, routes from a node and the sink are updated. Nodes have unlimited data buffers, because we want to study protocol performance when density varies without application-dependent constraints.

We focus on single monitoring traffic ($M = 1$), nodes generate packets with $r_m \in [0.01, .., 1]$. In a given simulation run, PGR is the same for all nodes, but nodes generate packets at some random instants.

The parameters for DA-MAC are the following : $t_l = 25\,ms$, interval between two channel polling of $250\,ms$ and $P_t = 0.95$. Each node estimates the number of neighbors within the radio range [74]. Both contention windows of SCP-MAC are of 16 slots, the polling period is $3\,ms$, the tone duration is $2\,ms$. SCP-MAC uses RTS and CTS messages, up to 3 re-transmissions per data packet and the Adaptive Channel Polling mechanism with parameter n equal to 3 [17]. The contention window of B-MAC is 32 slots. B-MAC and SCP-MAC have the same wake-up up period of $250\,ms$. Details about the system model and performance criteria description can be found in Sec. 3.3.

5.6.2 Numerical Results

We organize simulations in two scenarios. The first “Dynamic Network with Variable Node Density” focuses on analyzing the performance of the proposed DA-MAC in the dynamic scenario defined in Sec. 5.3. The latter, “Constant High Density Network” aims at illustrating how a high degree of density may impact network performance in different situations of traffic load.

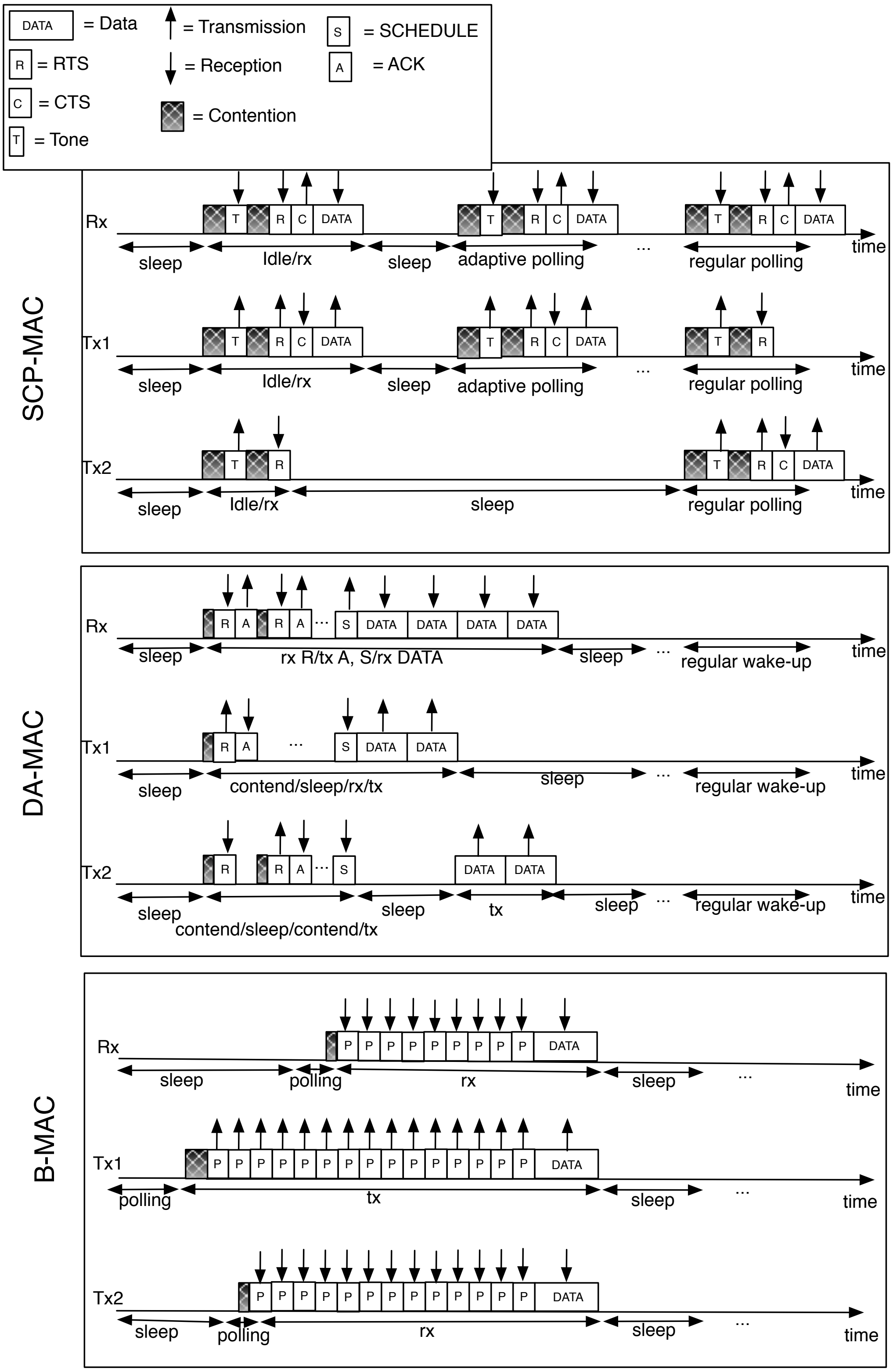


Figure 5.20 – Operation comparison of SCP-MAC, DA-MAC and B-MAC.

5.6.2.1 Scenario 1 : Dynamic Network with Variable Node Density

In this scenario, the network is composed of a *core sub-network* with a constant number of sensor nodes and a *dynamic sub-network* that goes through phases of deployment and extinction, which corresponds to time-variable node density.

At $t = 0\,s$, the network is composed of 30 core nodes with the average density per node of $N = 3.77$ neighbors. The network constructs routing tables and during 500 s, nodes belonging to the dynamic sub-network are progressively deployed until reaching the total network size of 50 with the average density of $N = 6.28$ neighbors. At $t_1 = 500\ s$, the deployment is finished and the network size is constant until $t_2 = 2500s$. At t_2, nodes of the dynamic sub-network begin to die until $t_3 = 3000\ s$. Between t_3 and $t_4 = 5000\ s$, the network is again composed of only core nodes. At t_4, a new deployment phase begins—the density instantly passes from $N = 3.77$ to $N = 12.57$. Until $t_5 = 7000\ s$, the network is composed of 100 nodes, then all dynamic nodes disappear instantaneously causing the density to drop again to the value of $N = 3.77$ and network size is constant until the end of the simulation.

Fig. 5.21 shows the average latency per node over the simulation time for r_m=0.1 *pps* (the average of 10 runs). We can see that the latency of DA-MAC exhibits constant performance for constant density—its parameters in the both operational periods (from t_1 to t_2) and (from t_4 to t_5) are result in constant latency. The figure also shows that even if SCP-MAC has been proven to be very efficient in 1-hop networks [17], it is outperformed by B-MAC with CW in high density multi-hop networks, because most of RTS packets collide in such networks, so nodes do not receive CTSs and backlogged packets must wait. Moreover, when average density increases, the Adaptive Channel Polling mechanism of SCP-MAC should adapt its parameter n (the number of extra channel polling) as a consequence, otherwise even three re-transmissions cannot guarantee a high delivery ratio. We note that with a realistic assumption of limited buffer size, most of the messages would be dropped instead of waiting unlimited time before being transmitted.

Fig. 5.22 shows the latency average per node over the simulation time for r_m=0.01 *pps*. We observe that when new nodes are deployed (t_4), we can observe that latency for SCP-MAC decreases, because new paths towards the sink appear and local congestion conditions are alleviated for some time. Unfortunately, when the new nodes disappear again, nodes return back to the same congestion conditions.

Fig. 5.23 shows the latency that results with a very heavy traffic load ($r_m = 1$ *pps*). Provided that traffic pattern is convergecast (all messages must be relayed to the same sink), nodes that are closer to the sink must face a very high congestion resulting in very high network latency.

Fig. 5.24 presents the average packet delivery ratio for the same dynamic network scenario with a varying packet generation rate. DA-MAC does not suffer from the congestion problem of SCP-MAC and B-MAC—even if latency increases due to the presence of new nodes, almost all packets are correctly received by the sink even for high traffic load.

Average energy consumption at the end of the simulation is illustrated in Fig. 5.25. Increasing traffic load results in higher energy consumption for all protocols. We observe that SCP-MAC results to be less energy consuming than other protocols thanks to the very low duty cycle. Nevertheless, low energy consumption is obtained at the expenses of very high latency and decreasing delivery ratio. DA-MAC shows energy consumption almost independent of traffic load, whereas in B-MAC high traffic load jeopardizes energy consumption.

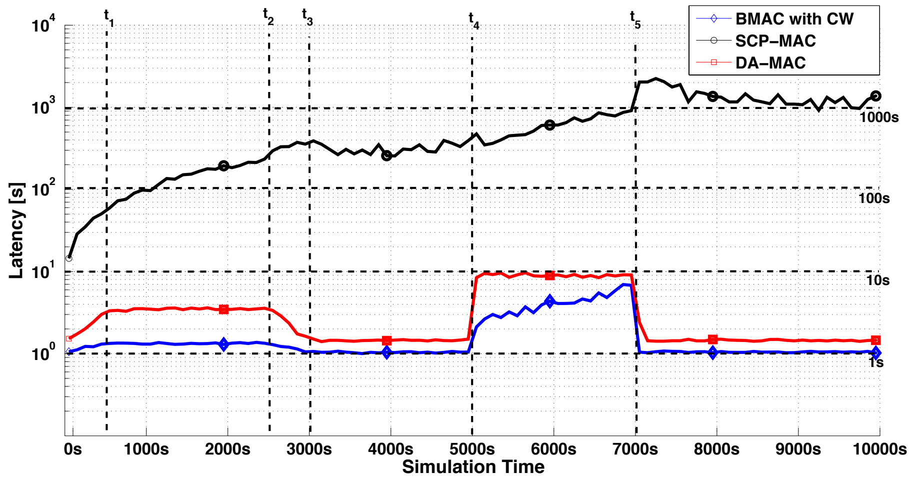


FIGURE 5.21 – Scenario 1, dynamic network with variable node density. Average Latency per node vs. simulation time. Packet generation rate is $r_m = 0.1$ *pps*.

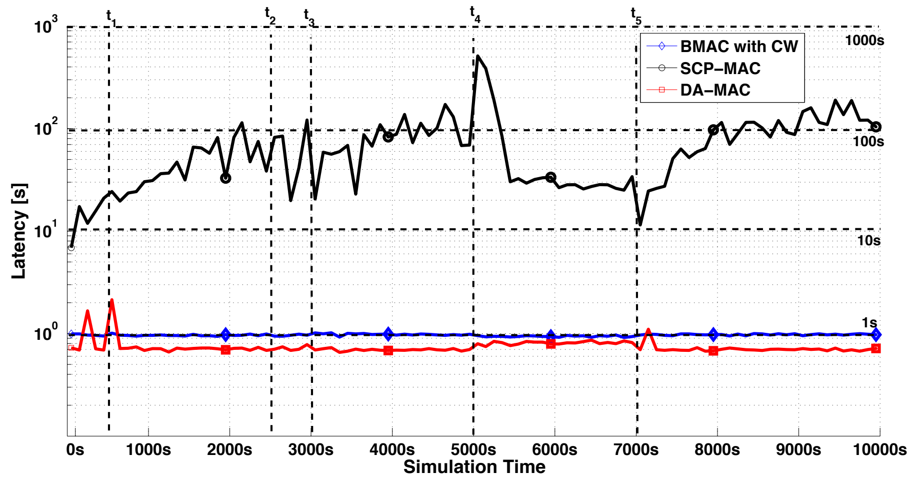


FIGURE 5.22 – Scenario 1, dynamic network with variable node density. Average Latency per node vs. simulation time. Packet generation rate is $r_m = 0.01$ *pps*.

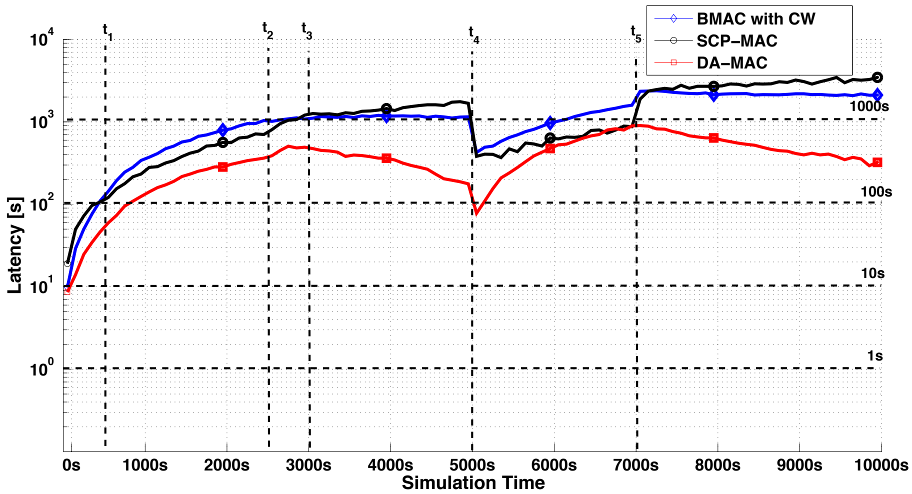


FIGURE 5.23 – Scenario 1, dynamic network with variable node density. Average Latency per node vs. simulation time. Packet generation rate is $r_m = 1\ pps$.

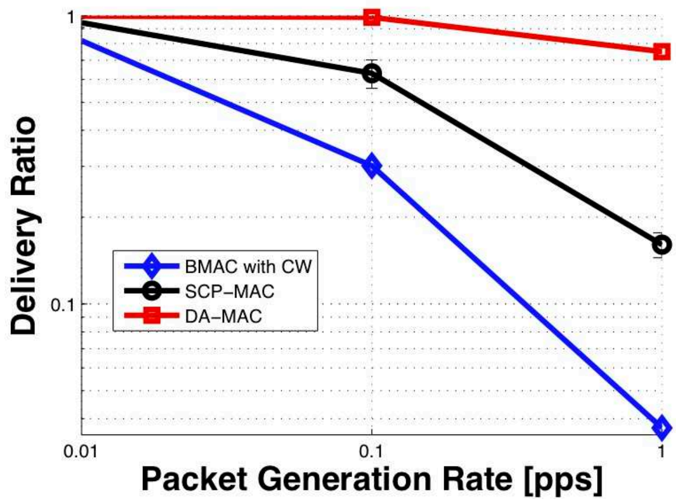


FIGURE 5.24 – Scenario 1, dynamic network with variable node density. Packet delivery ratio vs traffic load.

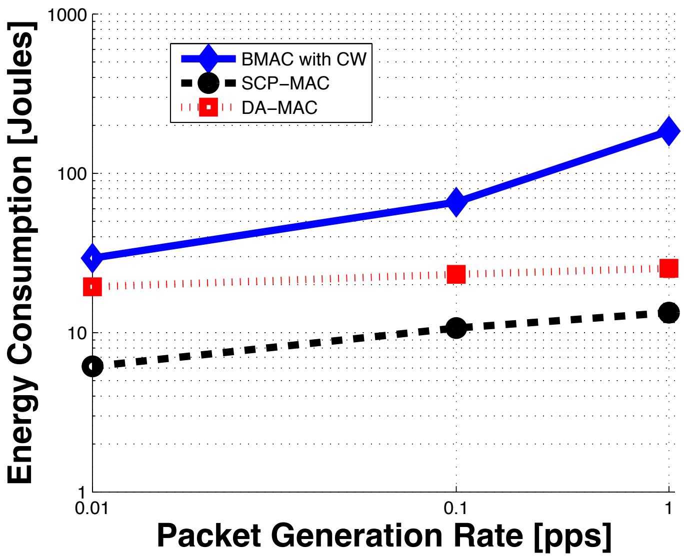


Figure 5.25 – Scenario 1, dynamic network with variable node density. Average energy consumption per node vs. traffic load.

5.6.2.2 Scenario 2 : Constant High Density Network

We have compared the MAC performance for a dense to very dense sensor network with constant density : three different networks with average densities of D = 3.77, 6.28, and 12.57 for network sizes of N =30, 50, and 100, respectively.

Figs. 5.26-5.28 show the average delivery ratio for light traffic (r_m=*0.01pps*) to very intensive traffic (r_m=*1pps*). Simulations show that DA-MAC outperforms other protocols for all conditions of traffic load and density : the delivery ratio is almost always the highest and around 100 % except in the case of a very loaded network. As expected, all protocols obtain better performance for lower values of r_m. As observed in the first scenario, with a multi-hop and dense network, B-MAC outperforms SCP-MAC in terms of delivery ratio. Even though density is constant, the delivery ratio of SCP is always lower than B-MAC and their performance become comparable in case of high traffic load and high density.

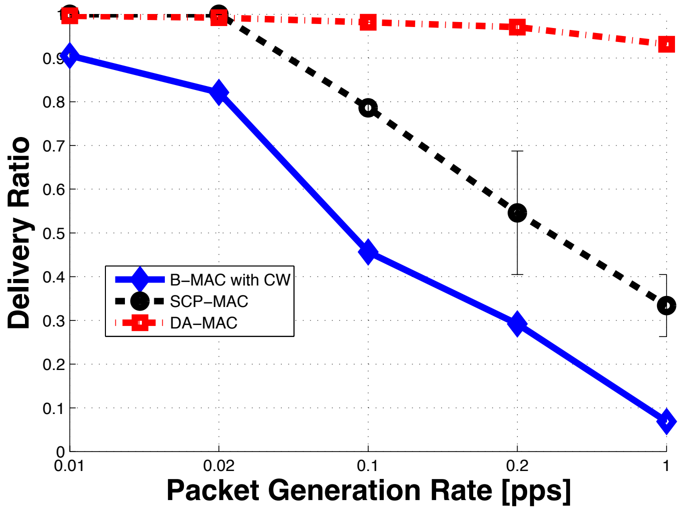


Figure 5.26 – Scenario 2, constant high density network. Average packet delivery ratio vs.traffic load per network size of 30 nodes.

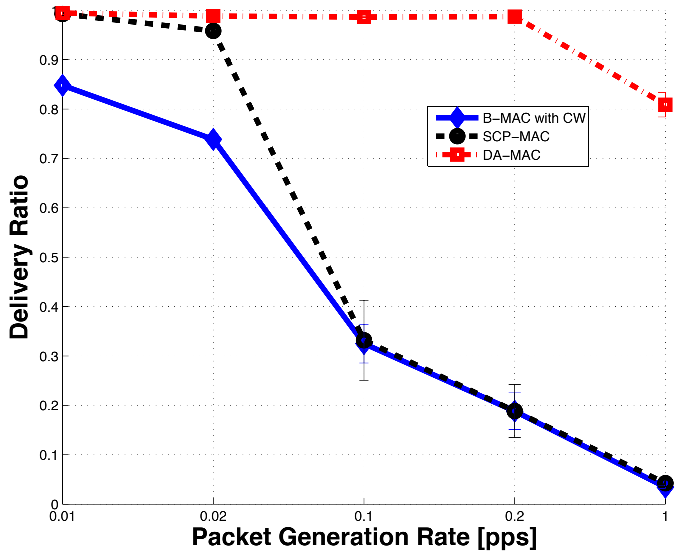


Figure 5.27 – Scenario 2, constant high density network. Average packet delivery ratio vs.traffic load per network size of 50 nodes.

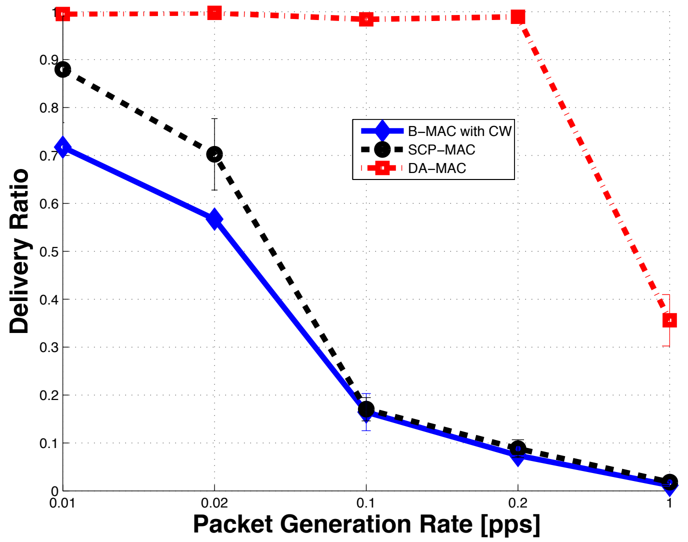


Figure 5.28 – Scenario 2, constant high density network. Average packet delivery ratio vs.traffic load per network size of 100 nodes.

Fig. 5.29-5.31 show the average energy consumption per node for different values of PGR. The energy consumption of DA-MAC is almost constant and independent of PGR variation. For B-MAC, energy consumption increases with increasing PGR. In SCP-MAC, when traffic is heavy, RTS collide very likely and nodes re-transmit packets very often. In B-MAC, when traffic is heavy, nodes spend most of the time transmitting preambles prior to data, which consumes energy. When traffic is lighter, they have less packets to send and spend more time in the sleeping state.

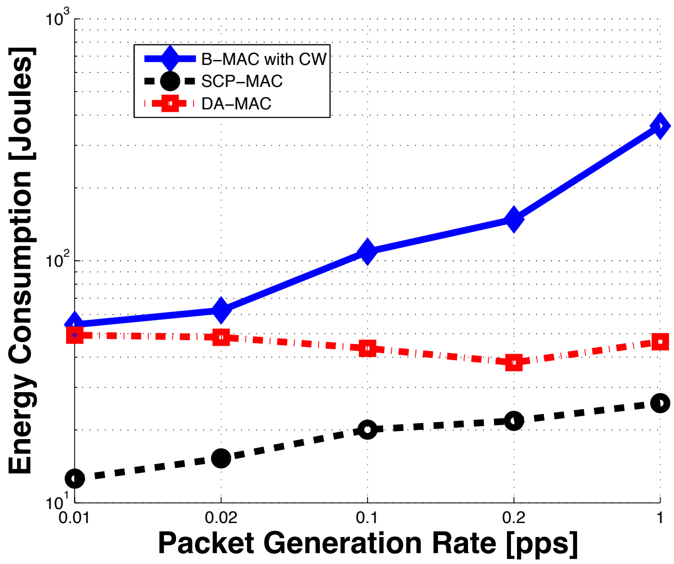


Figure 5.29 – Scenario 2, constant high density network. Average consumed energy per node vs. traffic load per network size of 30 nodes.

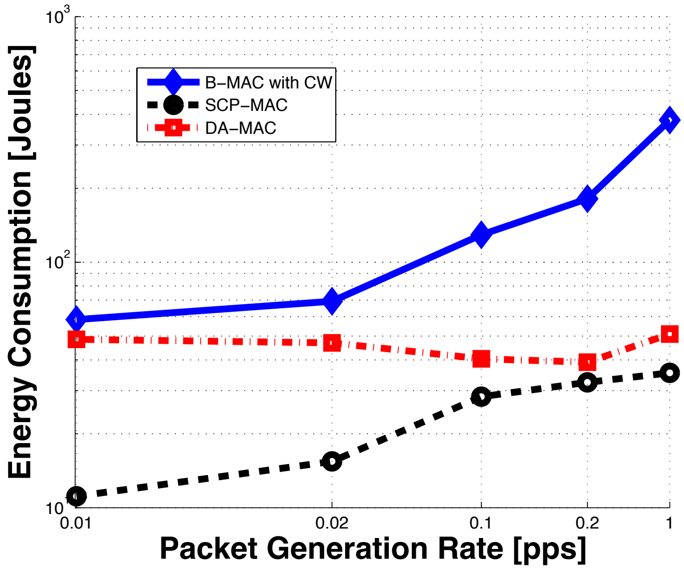


Figure 5.30 – Scenario 2, constant high density network. Average consumed energy per node vs. traffic load per network size of 50 nodes.

Figs. 5.32-5.34 show the average latency for different values of PGR. We can observe that for complex scenarios with high node density, SCP-MAC is outperformed by both DA-MAC and B-MAC. B-MAC and DA-MAC result in similar values of latency, however the same latency must be weighted by the fact that with DA-MAC delivery ration is always higher meaning that much more messages are delivered with the same average latency.

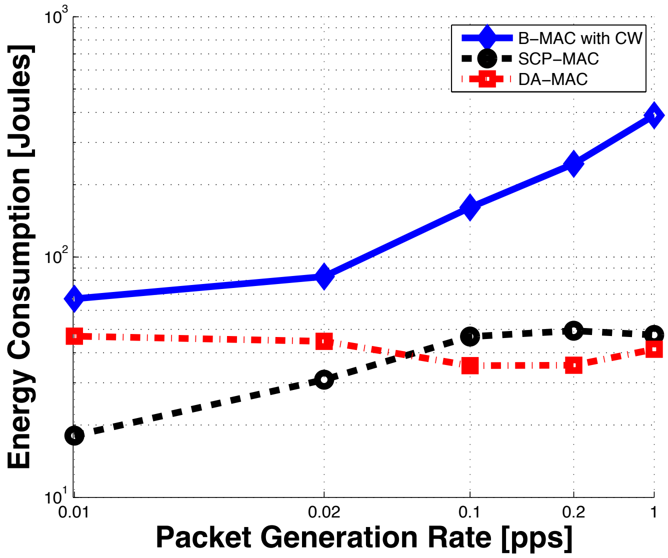


Figure 5.31 – Scenario 2, constant high density network. Average consumed energy per node vs. traffic load per network size of 100 nodes.

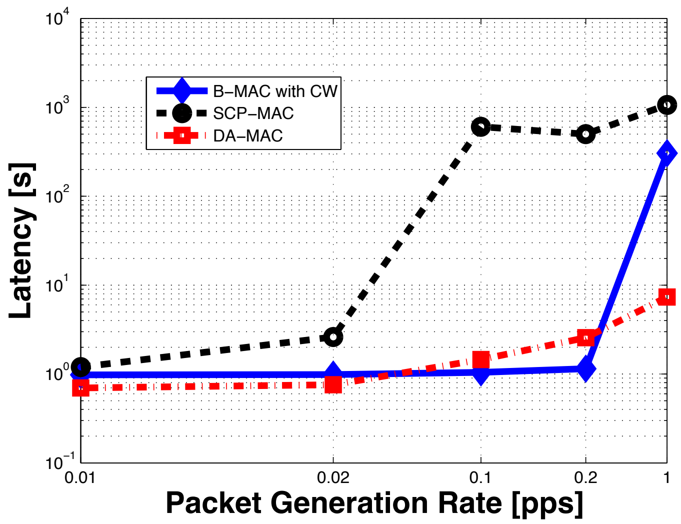


Figure 5.32 – Scenario 2, constant high density network. Average latency vs. traffic load per a network size of 30 nodes.

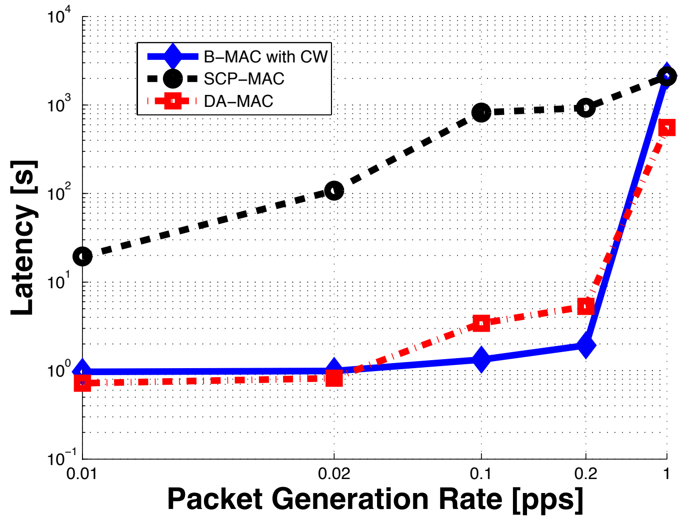


Figure 5.33 – Scenario 2, constant high density network. Average latency vs. traffic load per a network size of 50 nodes.

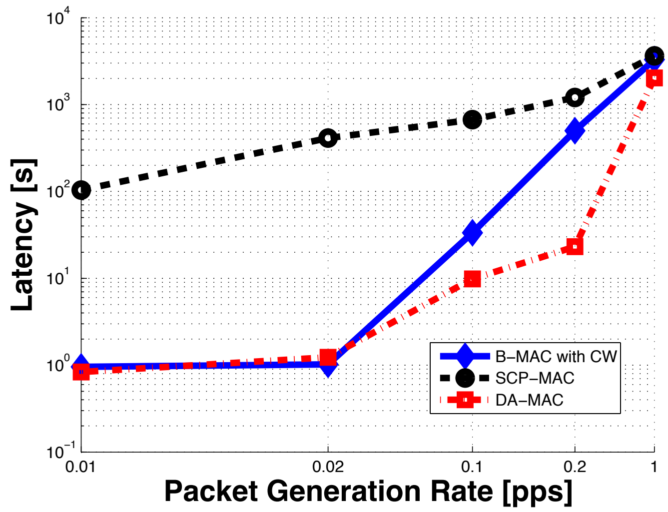


Figure 5.34 – Scenario 2, constant high density network. Average latency vs. traffic load per a network size of 100 nodes.

5.7 Conclusions

In dynamic sensor networks, each node have an average number of neighbors that is heterogeneously distributed over the space and that varies in time. In this chapter, we first dealt with the problem of channel access with contention window showing how the random distribution used to choose contention slots influences success probability. We came out with an adaptive contention window rule that depends upon local density of nodes aimed at achieving low collision probability. Second, we addressed the problem of density heterogeneity at MAC layer and we came out with a proposed density aware protocol that exploits the adaptive contention window rule : DA-MAC, an access method designed for sensor networks with time-varying number of nodes. Its principle is to offer a configurable channel sensing phase during which nodes request transmission opportunity in a way that avoids collisions. The receiver can thus schedule transmissions so that nodes may return to sleep and only wake up at their scheduled transmission instants. Allowing burst transmissions improves network capacity and the network can handle load fluctuations.

Our simulations concern both dynamic scenario with varying density and a constant density scenario with variable traffic load. In both scenarios we show that DA-MAC offers excellent performance in terms of packet delivery ratio and energy consumption. The comparisons with B-MAC and SCP-MAC show that under dynamic topology and heavy traffic, DA-MAC provides lower latency and high packet delivery ratio while consuming reasonable levels of energy. In the current implementation of DA-MAC, nodes apply the adaptive contention window using the uniform distribution for slot allocation. Coupling traffic differentiation using differentiated random distributions depending on traffic type as explained in the first part of this chapter, is straightforward thanks to the flexibility of DA-MAC and is left to future investigations.

Chapter 6

Conclusions and Future Perspective

6.1 Conclusions

The main contribution of this thesis is to investigate the problem of efficiency of MAC schemes in heterogeneous WSNs. We focused our attention on energy efficient protocols and investigated two major types of heterogeneous sensor networks : traffic-heterogeneous networks consisting of multiple sources with different characteristics and requirements, and density-heterogeneous networks consisting of different local regions with different levels of density of nodes.

In Chapter 2 of this dissertation, we provide a critical analysis of the most relevant MAC protocols for WSN present in the literature. We have investigated main contributions of each technique providing a critic analysis of their strong aspects and drawbacks. One of the leading concepts that have inspired our proposals is *preamble sampling*. This approach in fact is energy efficient, flexible, and easy to use, all essential requirements for operating in dynamic Wireless Sensor Networks.

In Chapter 3 of this dissertation we investigate the first kind of heterogeneity : traffic-heterogeneous networks. In such a network, nodes need to send different kinds of data with different priorities to one or multiple sinks. If the network is large, messages must be forwarded several times before reaching the final destination ; the result is that different relay-nodes across the network, which are also traffic sources, have different traffic loads with different priorities to handle. An inefficient MAC protocol may lead to high energy consumption without matching QoS requirements of different traffic. To cope with this problem, we proposed an adaptive MAC protocol, LA-MAC, for asynchronous sensor networks in which relay-nodes organize local transmissions of senders in a collision free manner to meet QoS requirements. Moreover, the protocol can absorb traffic load fluctuations thanks to the transmission of bursts. Extensive simulations that compare LA-MAC with B-MAC and X-MAC, two representative methods based on preamble sampling, show the efficiency of the proposed protocols in several scenarios with respect to latency, delivery ratio, and consumed energy.

In Chapter 4, we analyzed the energy consumption of preamble sampling MAC protocols by means of simple probabilistic modeling. We provided an analytic evaluation approach based on the instantaneous local traffic load in contrast to other approaches based on the network-wide traffic generation rate of sensors. Numerical simulations validated the proposed analytic model. We used the model to compare the energy consumption of the classical MAC PS protocols such as B-MAC and X-MAC with LA-MAC. Our analysis confirms that X-MAC is more energy efficient than B-MAC and highlights the energy savings achievable with LA-MAC with respect to both B-MAC and X-MAC.

The problem of density-heterogeneous networks was the subject of Chapter 5. In density varying networks, nodes and/or radio links may appear and disappear over the time due to several reasons such as battery exhaustion, node mobility, network management operations, etc. The result is that density distribution across the network is heterogeneous with the risk degraded performance at all protocol layers. In the chapter, we first observed how allocating contention slots of a CW according to different probability distribution function influences collision probability. We came out with an adaptive CW rule that uses the uniform distribution function so that resulting probability of successful transmission is above a given application threshold. Second, we applied the CW adaptation rule to a novel MAC density aware protocol suitable for dynamic networks : DA-MAC. Nodes take the advantage of the local density (density is estimated) to adapt protocol parameters to achieve energy efficient transmissions. Our simulations concern both dynamic scenario with varying density and a constant density scenario with variable traffic load. In both scenarios, we show that DA-MAC outperforms B-MAC and SCP-MAC providing lower latency and high packet delivery ratio while consuming reasonable levels of energy.

6.2 Future Work

Coupling Traffic Priority and Adaptive Contention Windows

In Chapter 5, we have observed that collision probability that results from the use of contention windows not only changes with the size of the window, but also with the random distributions that nodes use to select their contention slots. In the chapter, along with the uniform distribution, we have focused the attention on two other distributions : the negative exponential distribution and the Beta distributions. Both distributions are flexible, in fact, as they depend on shape factors they can be used to differentiate types of traffic. We aim at defining a rule that couples the application layer priority and the random distribution to be used by each station to select its contention slot. Such a rule, can be applied coupled not only with random access MAC protocols such as LA-MAC or DA-MAC but also with other protocols that use contention windows to limit the collision probability.

Mobility of Nodes

Nowadays Wireless Sensor Networks (WSNs) are largely deployed and used in several different scenarios. They are low-cost, long-living, and can be used in almost any imaginable environment like wildlife observation, industrial plant monitoring, or enhancing vehicular communications. Even if modern WSNs are *adaptive* (able to cope with different environments by adapting some parameters), there exist specific scenarios that require a focused and effective network design : Mobile-WSNs (M-WSNs) require such an approach.

Mobility scenarios in WSN can be divided into two categories : *active mobility* and *passive mobility.*

Active mobility involves all scenarios in which network elements spontaneously move from a position to another one to achieve a specific goal such as better *targeting* (nodes can move for close target proximity) [90] [91]. *Passive mobility* includes all scenarios in which network elements experience mobility and are not able to control it [4]. We will focus on *passive mobility* since we are interested in studying the effects of mobility onto network dynamics.

As observed in Chapter 5, mobility of nodes is one of the reasons that lead to density variations.

However, mobility of nodes has also effects over multiple elements of nodes architecture as Application, Networking and MAC related functions [92].
Major mobility-related issues in WSN are :

[Mobility of nodes] Position of sensor nodes may change over time. Localization algorithms must continuously feed mobility *estimation* algorithms for MAC adaptation. Another possibility is to simply *detect* mobility of nodes.

[Disconnection from neighbors] Mobile nodes must quickly connect to new neighbors.

[Energy waste] Mobility-awareness at the MAC layer is necessary to avoid energy waste because of turning the radio on when it is useless.

[Transmission during a movement] If a mobile node transmits while joining a new cluster, it may cause collisions if other transmissions occur at the same time.

[Changes in network topology] Mobile nodes cause network topology to change. Routes to the sinks must be updated or definitely recreated, otherwise packet loss may occur.

[Increasing traffic load] As a consequence of network topology change, network traffic load increases because of an increasing number of routing packets that must be sent.

[Redundancy of data] WSN applications must add position of the sensed quantity in each data packet. If the network was able to estimate mobility information of nodes such as future positions or trajectories, transmission and forwarding of redundant packets can be avoided. Current MAC protocols that support mobility differ with respect to the way they include mobility in their algorithms : *mobility is detected* or *mobility is predicted*.

It is our belief that by estimating the mobility of nodes to adapt some parameters at the MAC layer it is possible to achieve efficient mobility support. For example, if a central stations is able to estimate the mobility of different groups of nodes, it could predict the level of density of a particular region with a precision that depends upon the precision of density estimation. Then, future local density information can be used as an input to a density aware MAC such as DA-MAC.

Another alternative is to use distributed mobility estimation to feed a cross-layer MAC/Routing module to support mobility of nodes. In this case, the network is divided into reduced function mobile stations, that are not able to estimate their own mobility and full function nodes that are static and can estimate mobility of others. When a mobile node that needs to transmit data and a static node meet, the second one estimates the mobility of the first one and allocates transmission resources depending on mobility information, destination of data, and application priority.

Annexe A

Overview of OMNeT++ Simulator

All numerical simulations were run using the OMNeT++ open source simulator [93]. OMNeT++ is an object-oriented network framework that is based on discrete events. Its flexible and modular architecture permits to develop *modules* that can be interconnected to compose complex networks. Strong aspect of OMNeT++ is the graphical user interface that is available for developing and debugging purposes. Such interface, is highly useful thanks to animations and a console on each module that help the developer to investigate its models with high precision. In our work, focused on wireless sensor networks we developed network modules starting from templates available in the Mobility Framework, a framework including libraries and modules for WSNs [94].

Modules that compose a generic node are shown in Fig. A.1. Excepting from the deployment module that is described below, node structure is not changed with respect to basic nodes available with the Mobility Framework. It consists of :

- a battery, whose initial capacity is decremented each time a node switches from a radio mode to another,
- the application layer that generates messages,
- a network module that runs routing protocol,
- a Network Interface Card (NIC) that includes the MAC protocol, the SNR evaluation module, the threshold decider for collision detection, and the radio module that handles radio mode switch,
- the blackboard, used for cross layer information exchange,
- the mobility module, that handles mobility of the node,
- and a simple Address Resolution Protocol (ARP).

A.1 Contributions

Along with minor contributions on several basic modules, major contributions are the following :

- Modification of Mobility Framework Physical Layer : creation of new radio mode *polling*.

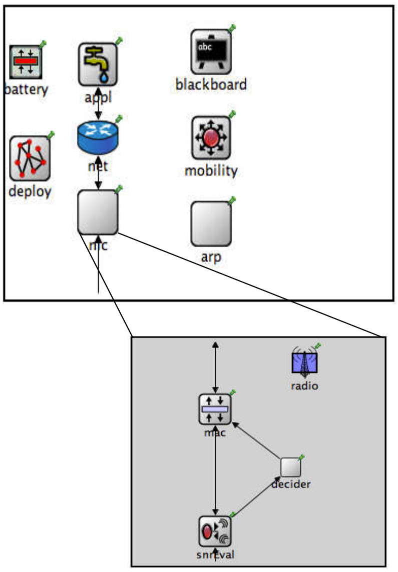


FIGURE A.1 – OMNeT++ modules composing a generic sensor node.

- Implementation of several MAC protocols : X-MAC, LA-MAC, DA-MAC, SCP-MAC.
- Implementation of Basic Operations of RPL Protocol.
- Implementation of Multi Traffic Application Layer.
- Implementation of Deployment Module to Create Dynamic Networks.

Creation of *Polling* Radio Mode

Mobility Framework came with a three modes radio able to Transmit, Receive and Sleep. However, when a node wakes up and polls the channel to detect some activity it remains in an idle mode that we called *polling mode*. Introducing an extra radio mode permits to study energy consumption more in details, in fact, energy consumption of idle mode is generally different from receiving mode and depends the specific radio chip-set that is used [86],[95],[96].

Basically, when a node wakes up, it automatically switches its radio into polling state, then if a radio frame is detected, radio turns to receiving mode. Provided that different radio modes consumes different energy, we also edited the Battery module in order to include the presence of a fourth mode.

Novel MAC protocols

In order to perform exhaustive simulations for comparison, several MAC protocols have been implemented. X-MAC and LA-MAC are based on preamble sampling approach. While DA-MAC and SCP-MAC are based on synchronous duty cycle. The Mobility Framework comes with an implementation of B-MAC protocol, a well known preamble sampling MAC. The implemented version of X-MAC is parametric with respect to the size of channel polling period and the desired initial duty cycle. Duration of gaps between two consecutive preambles depends on both radio bitrate and the size of ACK message. Duration of back-off for transmission as second user depends on both the size of data frame and bitrate.

Implementation of LA-MAC is more complex because parent node need to collect multiple preambles, clear them with early ACKs and send the SCHEDULE message at the end of its polling period. As in X-MAC, the user can choose the duration of channel polling duration and the initial duty cycle.

In the implemented version of SCP-MAC protocol, both contention windows are parametric. Also the number of extra channel polling is parametric and must be fixed off-line.

In DA-MAC nodes are able to learn the surrounding density by overhearing messages. To this end, when a frame is decoded and sent up to the MAC layer, the node reads the information embedded in the header before discarding the frame. As in X-MAC and LA-MAC, the user can choose the duration of channel polling duration and the initial duty cycle.

RPL Routing Protocol

All MAC protocol that we tested need the support of a routing protocol to provide the knowledge of the next hop address. Therefore we developed RPL (Routing Protocol for Low Power and Lossy Networks) [79]. Rpl defines the structure of a DAG (Directed Acyclic Graph) for Multi-Point-to-Point traffic (MP2P - routing packets to a single sink) and supports multiple sinks with parallel DAGs. In RPL, routes are built depending on the weight of each link, in our implementation, both ETX based weights [87] and random weights can be used.

Multi Traffic Application Layer

In order to test multiple parallel traffics, we modified the Sensor Application Layer class that comes with the Mobility Framework. We defined two classes of traffic, urgent and non-urgent. Priority of urgent traffic goes from 1 to 5, whereas priority of non-urgent messages goes from 6 to a value that is provided in the configuration file. The user can decide how many devices will generate whichever class of traffic. In particular when routing layer has defined the rank of all nodes with respect to each one of the sinks, the user can decide to let only one device that is a given distance from a sink to be the source of urgent messages. As explained in Sec. 3.3 urgent messages are characterized by bursts of frames that are generated with a given probability Γ. If there are multiple sinks, the user can decide if all sinks can receive all classes of traffic or if a sink is interested in a particular class of traffic.

Deployment Module

In order to test the efficiency of proposed DA-MAC protocol (cf. Chapter 5) we developed the Deployment Module that is responsible for dynamic deployment of nodes. The module is very flexible and the user can define a differentiated deployment pattern that each node must follow. Deployment pattern must be provided in the configuration file. The deployment pattern that we defined in Sec. **??** is an example of a pattern in 6 phases.

List of Publications

Conference Papers

[C1] G. Corbellini and E. Calvanese Strinati and E. Ben Hamida and A. Duda, “DA-MAC : Density Aware MAC for Dynamic Wireless Sensor Networks”, 22nd IEEE Personal Indoor Mobile Radio Communications (PIMRC’11 - LPAN), pp.920-924, Toronto, Canada, September 2011

[C2] G. Corbellini and E. Calvanese Strinati and A. Duda “LA-MAC : Low-Latency Asynchronous MAC for Wireless Sensor Networks”, accepted for publication at the 23rd IEEE Personal Indoor Mobile Radio Communications (PIMRC’12).

[C3] G. Corbellini and C. Abgrall and E. Calvanese Strinati and A. Duda “Energy Evaluation of Preamble Sampling MAC for Wireless Sensor Networks’, accepted for publication at the 23rd IEEE Personal Indoor Mobile Radio Communications (PIMRC’12).

Patents

[P1] G. Corbellini, E. Calvanese Strinati, "Méthode de Communication Asynchrone pour réseau des Capteurs Sans Fil," DD11772.

[P2] G. Corbellini, E. Calvanese Strinati, "Procédé d’Accès a un Canal de Transmission dans un Réseau de Communication Sans Fil à Fenêtre de Contention," DD12561.

Bibliographie

[1] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat Monitoring with Sensor Networks," *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[2] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," *in Proceedings of the 7th symposium on Operating systems design and implementation*, pp. 381–396, 2006.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking : Design Tradeoffs and Early Experiences with ZebraNet," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, pp. 96–107, 2002.

[4] "ARGO - Global Ocean Sensor Network." http://www.argo.ucsd.edu.

[5] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard Computing : Sensor Networks in Agricultural Production," *Pervasive Computing, IEEE*, vol. 3, no. 1, pp. 38–45, 2004.

[6] J. Elson and D. Estrin, "Sensor Networks : A Bridge to the Physical World," *Book chapter : Wireless Sensor Networks*, pp. 3–20, 2004.

[7] K. Langendoen, "Energy-Efficient Medium Access Control," *Book chapter : Medium Access Control in Wireless Networks, H. Wu and Y. Pan (editors), Nova Science Publishers*, 2008.

[8] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks," *in Proc. of 18th IEEE IPDPS*, p. 224, 2004.

[9] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient Collision-Free Medium Access Control for Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 181–192, 2003.

[10] Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. of Annual IEEE International Conference on Computer Communications, INFOCOM*, vol. 3, pp. 1567–1576, 2002.

[11] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 171–180, 2003.

[12] A. El-Hoiydi, "Aloha with Preamble Sampling for Sporadic Traffic in Ad Hoc Wireless Sensor Networks," *in Proc. of the Int. Conf. on Communications*, vol. 5, pp. 3418–3423, New York, NY, USA, April 2002.

[13] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 95–107, 2004.

[14] S. Mahlknecht and M. Boeck, "CSMA-MPS : A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks," *in Proc. of IEEE Workshop on Factory Communication Systems*, pp. 73–80, Vienna, Austria, September 2004.

[15] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC : a Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 307–320, 2006.

[16] C. Enz, A. El-Hoiydi, J. Decotignie, V. Peiris., "WiseNET : An Ultra-Low-Power Wireless Sensor Network Solution," *IEEE Computer Society Press*, vol. 37, pp. 62–70, August 2004.

[17] W. Ye, F. Silva and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 321–334, Boulder, CO, USA, November 2006.

[18] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *in ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 5, 2009.

[19] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks," *in Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 222–248, 2010.

[20] "ICT-258512 EXALTED project." http://www.ict-exalted.eu/.

[21] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks : a Survey," *in IEEE Communications Magazine*, vol. 44, pp. 115–21, April 2006.

[22] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC : a Low Latency, Energy Efficient Demand-wake-up MAC Protocol for Wireless Sensor Networks," *in Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc*, pp. 53–62, May Hong Kong, China, 2008.

[23] G. Halkes and K. Langendoen, "Crankshaft : An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks," *in Proc. of European Conference on Wireless Sensor Networks, EWSN*, pp. 228–244, 2007.

[24] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *in IEEE Transactions on Mobile Computing*, vol. 1, pp. 70–80, January-March 2002.

[25] C. Merlin and W. Heinzelman, "Network-aware Adaptation of MAC Scheduling for Wireless Sensor Networks," *in Proc. of the Conference on Distributed Computing in Sensor Systems (Poster Session)*, pp. 24–28, 2007.

[26] IEEE 802.15.4, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)." http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf, 2003.

[27] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," *in System Sciences. Proceedings of the 33rd Annual Hawaii International Conference on*, pp. 10–pp, 2000.

[28] K. Arisha, M. Youssef and M. Younis, "Energy-Aware TDMA-Based MAC for Sensor Networks," *in Proc. of IEEE IMPACCT*, pp. 21–40, New York City, NY, May 2002.

[29] S. Coleri-Ergen and P. Varaiya, "PEDAMACS : Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks," *in IEEE Transactions on Mobile Computing*, vol. 5, no. 7, pp. 920–30, 2006.

[30] S. Chatterjea, L. van Hoesel, P. Havinga, "AI-LMAC : An Adaptive, Information-centric and Lightweight MAC Protocol for Wireless Sensor Networks," *in Proc. of Intelligent Sensors, Sensor Networks and Information Processing Conference, ISSNIP*, 2004.

[31] L. van Hoesel, P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks : Reducing Preamble Transmissions and Transceiver State Switches," *in Proc. of IEEE INSS*, Tokyo, Japan 2004.

[32] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Energy-Efficient, Application-Aware Medium Access for Sensor Networks," *in Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*, pp. 8–pp, 2005.

[33] S. Mank, R. Karnapke, and J. Nolte, "An Adaptive TDMA based MAC Protocol for Mobile Wireless Sensor Networks," *in Proc. of Sensor Technologies and Applications, SensorComm. International Conference on*, pp. 62–69, 2007.

[34] B. Yahya and J. Ben-Othman, "An Adaptive Mobility Aware and Energy Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. of Computers and Communications, ISCC. IEEE Symposium on*, pp. 15 –21, July 2009.

[35] M. Ali, T. Suleman, and Z. Uzmi, "MMAC : A Mobility-Adaptive, Collision-Free MAC Protocol for Wireless Sensor Networks," *in Proc. of Performance, Computing, and Communications Conference, IPCCC. 24th IEEE International*, pp. 401–407, 2005.

[36] H. Cao, K. W. Parker, A. Arora, "O-MAC : A Receiver Centric Power Management Protocol," *in Proc. of IEEE ICNP*, pp. 311–320, Santa Barbara, CA, November 2006.

[37] M. Ringwald and K. Roemer, "BitMAC : A Deterministic, Collision-Free, and Robust MAC Protocol for Sensor Networks," *in Proc. of European Conference on Wireless Sensor Networks, EWSN*, pp. 57–69, 2005.

[38] G. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," *in Proc. of IEEE Military Communications Conference, MILCOM*, vol. 1, pp. 347–351, 2001.

[39] M. Brownfield, K. Mehrjoo, A. Fayez, and N. Davis IV, "Wireless Sensor Network Energy-Adaptive MAC Protocol," *in Proc. of IEEE CCNC*, 2006.

[40] J. Li and G. Lazarou, "A Bit-Map-Assisted Energy-Efficient MAC Scheme for Wireless Sensor Networks," *in Proc. of ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN*, pp. 55–60, 2004.

[41] L. van Hoesel, P. Havinga, "Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs," *in Proc. of the IEEE Vehicular Technology Conference, VTC*, vol. 3, pp. 1598–1602, 2004.

[42] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *in IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[43] S. Kulkarni and M. Arumugam, "SS-TDMA : A Self-Stabilizing MAC for Sensor Networks," *in Sensor Network Operations, IEEE Press*, 2006.

[44] Y. Kim, H. Shin, H. Cha, "Y-MAC : An Energy-efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks," *in Proc. of ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN*, pp. 53–63, St. Louis, Missouri, April 2008.

[45] S. D. O. Incel and P. Jansen, "Multi-channel Support for Dense Wireless Sensor Networking," *in Proc. of European Conference on Smart Sensing and Context, EuroSSC*, pp. 1–14, 2006.

[46] H. Pham and S. Jha, "An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC)," *in Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*, pp. 588–60, Fort Lauderdale, FL, October 2004.

[47] P. Raviraj, H. Sharif, M. Hempel, S. Ci, H. Ali, and J. Youn, "A Mobility Based Link Layer Approach for Mobile Wireless Sensor Networks," *in Proc. of Electro Information Technology, IEEE International Conference on*, pp. 1–6, May. 2005.

[48] S. Du, A. Saha, and D. Johnson, "RMAC : A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," *in Proc. of Annual IEEE International Conference on Computer Communications, INFOCOM*, pp. 1478–1486, 2007.

[49] W. Ye and J. Heidemann and D. Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks," *IEEE ACM Transactions on Networking*, vol. 12, pp. 493–506, June 2004.

[50] A. Woo and D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *in Proc. of IEEE/ACM International Conference on Mobile Computing and Networking, MobiCom*, pp. 221–235, Rome, Italy, July 2001.

[51] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio, "Increasing the Efficiency of Preamble Sampling Protocols for Wireless Sensor Networks," *in Proc. of the First Mobile Computing and Wireless Communication International Conference, MCWC*, pp. 117–122, 2006.

[52] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A Wireless Sensor MAC Protocol for Bursty Data Traffic," *in Proc. of IEEE Personal, Indoor Mobile Radio Communications Conf., PIMRC*, pp. 1–5, Athens, Greece, September 2007.

[53] J. Rabaey et al., "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking," *in IEEE Computer Magazine*, vol. 33, no. 7, pp. 42–48, 2000.

[54] C.-J. Liang, B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi Interference in Low Power ZigBee Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 309–322, Nov. 2010.

[55] A. Bachir, D. Barthel, M. Heusse and A. Duda, "Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks," *in Proc. of the Int. Conf. on Communications*, vol. 7, pp. 3365–3370, Istanbul, Turkey, June 2006.

[56] M. Miller and N. Vaidya, "A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio," *in IEEE Transactions on Mobile Computing*, vol. 4, pp. 228–42, May/June 2005.

[57] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC : a Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 1–14, 2008.

[58] P. Dutta, S. Dawson-Haggerty, Y. Chen, C. Liang, and A. Terzis, "Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless," *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 1–14, 2010.

[59] R. Kuntz and T. Noël, "Machiavel : Accessing the medium in mobile and dense WSN," *in Proc. of IEEE Personal, Indoor Mobile Radio Communications Conf., PIMRC*, pp. 1088–1092, 2009.

[60] R. Kuntz, A. Gallais, and T. Noel, "Auto-adaptive MAC for energy-efficient burst transmissions in wireless sensor networks," *in Proc. of IEEE Wireless Communications and Networking Conference, WCNC*, pp. 233 –238, march 2011.

[61] K. Jamieson, H. Balakrishnan, and Y. Tay, "Sift : A MAC Protocol for Event-driven Wireless Sensor Networks," *in Wireless Sensor Networks*, pp. 260–275, 2006.

[62] K. Wong and D. Arvind, "SpeckMAC : Low-power Decentralized MAC Protocols for Low Data Rate Transmissions in Specknets," *in Proc. of the 2nd International Workshop on Multi-hop Ad Hoc Networks : from Theory to Reality*, pp. 71–78, 2006.

[63] A. B. U. Roedig and C. Sreenan, "f-MAC : A Deterministic Media Access Control Protocol Without Time Synchronization," *in Proc. of European Conference on Wireless Sensor Networks, EWSN*, pp. 276–291, 2006.

[64] I. Rhee, A. Warrier, M. Aia, J. Min, and M. Sichitiu, "Z-MAC : a Hybrid MAC for Wireless Sensor Networks," *IEEE ACM Transactions on Networking*, vol. 16, pp. 511–524, San Diego, CA, November 2008.

[65] T. Zheng, S. Radhakrishnan and V. Sarangan, "PMAC : an Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *in Proc. of Parallel and Distributed Processing Symposium. 19th IEEE International*, pp. 8–pp, April 2005.

[66] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement. Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 2 : Higher-Speed Physical Layer (PHY) Extension in the 2.4 GHz Band - Corrigendum 1," *IEEE Std 802.11b-1999/Cor 1-2001*, pp. 0–1, 2001.

[67] Z. Zaidi and B. Mark, "Mobility Estimation for Wireless Networks Based on an Autoregressive Model," *in Proc. of Global Telecommunications Conference, GLOBECOM*, vol. 6, pp. 3405 – 3409 Vol.6, nov. 2004.

[68] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *in IEEE Transactions on Wireless Communications*, vol. 1, pp. 660 – 670, oct. 2002.

[69] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense : an Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless Lans," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 121–132, 2005.

[70] Z. Abichar and J. Chang, "CONTI : Constant-Time Contention Resolution for WLAN Access," *in NETWORKING. Networking Technologies, Services, and Protocols ; Performance of Computer and Communication Networks ; Mobile and Wireless Communications Systems*, pp. 263–287, 2005.

[71] J. Galtier, "Analysis and Optimization of MAC with Constant Size Congestion Window for WLAN," *in Proc. of Systems and Networks Communications, ICSNC. Second International Conference on*, p. 25, 2007.

[72] J. Hill and D. Culler, "Mica : a Wireless Platform for Deeply Embedded Networks," *in Micro, IEEE*, vol. 22, no. 6, pp. 12–24, 2002.

[73] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND : Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks," *in Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc*, pp. 190–201, Florence, Italy, May 2006.

[74] M.-H. Chiang and G. Byrd, "Neighborhood-Aware Density Control in Wireless Sensor Networks," *in Proc. of Sensor Networks, Ubiquitous and Trustworthy Computing, SUTC. IEEE International Conference on*, pp. 122 –129, 2008.

[75] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive Energy-conserving Routing for Multihop Ad Hoc Networks," *in Research Report 527, USC/Information Sciences Institute*, 2000.

[76] X. Wu, G. Chen, and S. Das, "On the Energy Hole Problem of Nonuniform Node Distribution in Wireless Sensor Networks," *in Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*, pp. 180–187, 2007.

[77] J. Polastre, "A Unifying Link Abstraction for Wireless Sensor Networks," *PhD Thesis*, University of California, Berkeley October 2005.

[78] J. Steffan, L. Fiege, M. Cilia, and A. Buchmann, “Towards Multi-Purpose Wireless Sensor Networks,” *in Proc. of International Conference on Sensor Networks, SENET*, August 2005.

[79] The ROLL Design Team, “RPL : Routing Protocol for Low Power and Lossy Networks.” https://datatracker.ietf.org/doc/draft-ietf-roll-rpl/, March 2011.

[80] A. Bachir, S. Plancoulaine, D. Barthel, M. Heusse, and A. Duda, “Frame Preamble MAC for Multihop Wireless Sensor Networks : Design and Implementation,” *in Ad Hoc and Wireless Sensor Networks, Old City Publishing*, 2008.

[81] “Heterogeneity and Homogeneity.” http://en.wikipedia.org/wiki/Homogeneity_and_heterogeneity, 2012.

[82] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, “Exploiting Heterogeneity in Sensor Networks,” *in Proc. of Annual IEEE International Conference on Computer Communications, INFOCOM*, vol. 2, pp. 878–890 vol. 2, 2005.

[83] X. Du, Y. Xiao, and F. Dai, “Increasing Network Lifetime by Balancing Node Energy Consumption in Heterogeneous Sensor Networks : Research Articles,” *in Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 125–136, 2008.

[84] M. Mathis, J. Mahdavi, S. Floyd, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options.” RFC 2018, 1996.

[85] R. Kuntz, A. Gallais, and T. Noël, “From Versatility to Auto-Adaptation of the Medium Access Control in Wireless Sensor Networks,” *in Journal of Parallel and Distributed Computing*, vol. 71, no. 9, pp. 1236–1248, 2011.

[86] “Texas Instruments, CC1100 datasheet.” http://focus.ti.com/docs/prod/folders/print/cc1100.html.

[87] D. De Couto, “High-Throughput Routing for Multi-Hop Wireless Networks.” http://pdos.csail.mit.edu/papers/grid:decouto-phd/thesis.pdf, 2004.

[88] The ROLL Design Team, “Routing Metrics used for Path Calculation in Low Power and Lossy Networks.” https://datatracker.ietf.org/doc/draft-ietf-roll-routing-metrics/, March 2011.

[89] C. Wan, S. Eisenman, A. Campbell, and J. Crowcroft, “Siphon : Overload Traffic Management Using Multi-radio Virtual Sinks in Sensor Networks,” *in Proc of. ACM International Conference on Embedded Networked Sensor Systems, SenSys*, pp. 116–129, 2005.

[90] S. Munir, B. Ren, W. Jiao, B. Wang, D. Xie, and M. Ma, “Mobile Wireless Sensor Network : Architecture and Enabling Technologies for Ubiquitous Computing,” *in Proc. of Advanced Information Networking and Applications Workshops, AINAW.*, vol. 2, pp. 113 –120, may. 2007.

[91] M. Zhang, X. Du, and K. Nygard, “Improving Coverage Performance in Sensor Networks by Using Mobile Sensors,” *in Proc. of IEEE Military Communications Conference, MILCOM*, vol. 5, pp. 3335–3341, 2005.

[92] A. Raja and X. Su, “Mobility handling in MAC for wireless ad hoc networks,” vol. 9, no. 3, pp. 303–311, 2009.

[93] “OMNeT++ Discrete Event Simulator.” http://www.omnetpp.org.

[94] “Mobility Framework for OMNeT++.” http://mobility-fw.sourceforge.net.

[95] C. Corporation, “CC2420 Single Chip Low Cost Low Power RF Transceiver, Data Sheet.” http://www.ti.com/product/cc2420.

[96] C. Corporation, “CC2500 Single Chip Low Cost Low Power RF Transceiver, Data Sheet.” http://www.ti.com/product/cc2500.